PMIExplorer

Configuration and user manual

2.0.7

Copyright © 2014-2015 by PMIExplorer.com

Table of Contents

About PMIExplorer....4
Quick start guide....5
  Remote agent....5
  Clients....6
  User interface....6
Installation and preparation....8
  Topology....8
  Installing PMIExplorer....9
  Java runtime environment....10
  Preparing the database....11
  Preparing security....15
  Importing the license....15
Configuration....17
  Configuration files....17
  User interface....17
  PMI Collector....18
  Database connectivity....20
  LDAP Security....24
  Extension points....26
  Remote agent....27
  Notifications....28
  Logging....30
Running PMIExplorer....31
  Starting the client....31
  Starting the remote agent....31
  Stopping the remote agent....32
  Running as a service....32
User interface....33
  Layout....33
  Workspaces....34
  PMI metrics tree....35
  Charting....37
  Detail panels....41
  System tray....44
Connecting to WebSphere....46
  Connection context....46
  Creating the connection....46
  Common connection problems....48
PMI data collector....50
  PMI data in PMIExplorer....50
  Sizing the database....52
  Monitoring metrics....54
  Monitoring profiles....55
  Data maintenance....56
  Exporting data....58

Alerting....60
Alerting mechanisms....60
Areas of interest....62
Managing alarms....63
Defining alerts....65
Managing licenses....69
About the licensing system....69
Importing the license key....70
Security....72
Authentication....72
Available roles....73
Authorization....73
Setting up the keystore....75
Remote agent....76
Connecting to the remote agent....76
Communication....77
Connection security....78
Persistent monitoring....78

About PMIExplorer

PMIExplorer has been designed to get more value out of your WebSphere PMI data. A standalone client or remote agent monitors your WebSphere performance and is able to record and combine data from multiple servers while providing a great level of detail. PMIExplorer provides a complete, affordable and high quality solution to analyze the performance of your environments and sends alerts when metrics exceed configurable thresholds. Use PMIExplorer for performance- and capacity management or troubleshooting.

The power and added value of PMIExplorer comes from additional computations on the collected data. PMIExplorer calculates the actual responsetimes between consecutive samples instead of presenting an average since server startup. These delta computations also provide additional insight into other areas which you can explore with PMIExplorer.

PMIExplorer also comes with an API which enables you to monitor the performance of any function in your application by adding the @Instrument annotation in your Java code. The PMIExplorer API takes care of registering a PMI metric in WebSphere Application Server, which enables you to use these statistics for performance analysis or alerting.

PMIExplorer monitors your WebSphere environments across the boundaries of a JVM and combines data from multiple servers which might be in different cells. The collector component can run as a remote agent which keeps collecting data in the background. The collector depends on an AdminClient connection and leverages the WebSphere security model to access the PMI data. PMIExplorer additionally adds role based access control to protect your data and monitoring- and alerting configuration.

Quick start guide

Remote agent

Step 1: Extract the archive on your preferred location

PMIExplorer can be installed in any location. You can choose for co-location with your WebSphere server or deployment manager and share the Java runtime. It is also possible to run PMIExplorer on a dedicated machine.

Step 2: Ensure proper Java version is available

PMIExplorer requires IBM Java 7 to connect to the WebSphere environments. IBM provides downloads on their developerworks website: http://www.ibm.com/developerworks/java/jdk.

Step 3: Add the required WebSphere libraries

Copy the `com.ibm.ws.admin.client-x.x.x.jar` and `com.ibm.ws.security.crypto.jar` libraries to the PMIExplorer `ext` directory. All libraries in `ext` directory are loaded during PMIExplorer startup.

Step 4: Configure the remote agent

Open the configuration file `config/PMIExplorer.conf` and set the values for the `pmi.agent.*` properties. The property `pmi.agent.enabled` needs to be set to true for the agent to start.

Step 5: Configure database

This step is optional but recommended. Copy your database JDBC drivers to the `ext` directory and modify the `pmi.jdbc.*` properties to configure the connection. PMIExplorer can automatically generate the required tables and indexes or a script file to run manually.

Step 6: Configure security

This step is optional. Configure the `pmi.ldap.*` properties to connect to your directory server. You might need to add your certificate to the PMIExplorer keystore. Once connected with the client you need to link your LDAP users and groups to the available security roles.

Step 7: Load the license key

Refer to chapter `Managing licenses` on page `69` for details on loading the license key for the remote agent. The PMIExplorer remote agent will take care of distributing the available license keys to the connected clients.

Step 8: Connect the clients

Setup the clients. You also might want to set the `pmi.agent.host, pmi.agent.port` and

`pmi.agent.workspaces` properties on the clients to make the them aware of the agents location and workspaces before they connect to the agent.

Step 9: Start exploring

You can now connect to your WebSphere environment(s), define your monitoring configuration and set up alerts.

Clients

Step 1: Extract the archive on your preferred location

PMIExplorer can be installed in any location. Extract the archive to your favorite location on your laptop or desktop.

Step 2: Ensure proper Java version is available

PMIExplorer requires a Java 7 runtime to operate. IBM Java 7 is required to connect to WebSphere environments, otherwise the Oracle Java 7 runtime will suffice. IBM Java 7 is available from developerworks: http://www.ibm.com/developerworks/java/jdk.

Step 3: Optionally add the WebSphere libraries

The WebSphere libraries are required when you want to connect the client directly to WebSphere environments. Copy the `com.ibm.ws.admin.client-x.x.x.jar` and `com.ibm.ws.security.crypto.jar` libraries from your WebSphere installation to the PMIExplorer `ext` directory.

Step 4: Load the license key

Refer to chapter `Managing licenses` on page `69` for details on loading the license key for the standalone client. In a remote agent setup the client will automatically retrieve the license key from the remote agent when you connect for the first time.

Step 5: Start exploring

You can now connect to your agent or WebSphere environment(s). Define your monitoring configuration, set up alerts and start exploring.

User interface

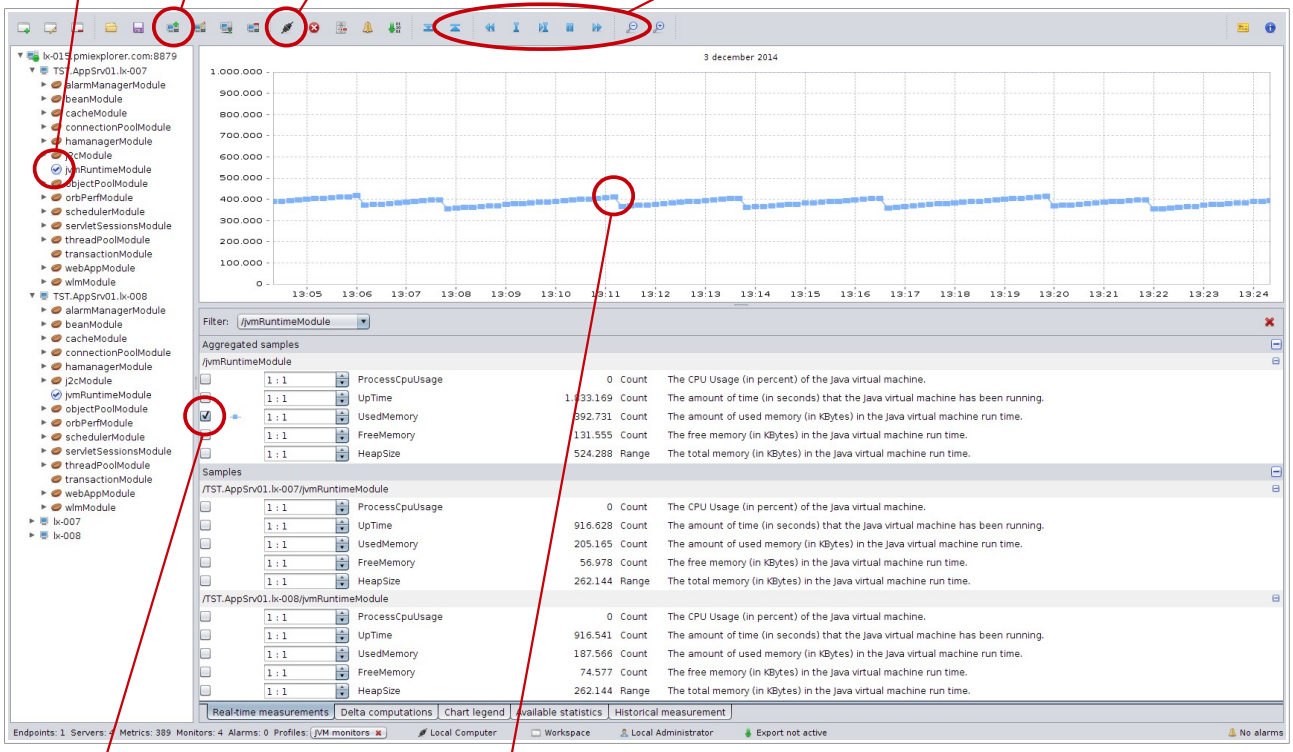
The image on the next page show the most important buttons to get you started quickly in the PMIExplorer user interface.

3. Start monitoring a metric

2. Connect to a WebSphere endpoint

1. Connect to the remote agent

5. Work with chart controls



6. Click on a data point to view recorded details

4. Check to plot a statistic on the chart

Installation and preparation

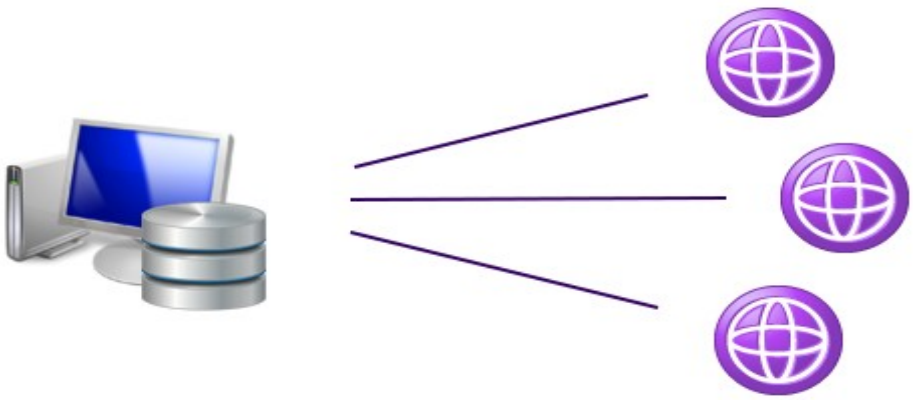
Topology

PMIExplorer can be installed in two possible topologies. The details of these configurations are described in this paragraph. For both topologies these constraints apply:

1. The database cannot be shared with other instances of PMIExplorer
2. The user repository (LDAP) can be shared, but it's also possible to use multiple repositories for multiple instances of PMIExplorer
3. Multiple PMIExplorer instances can connect to the same WebSphere endpoint, but that would result in additional load on the endpoint and possibly unexpected values in the case of CPU utilization where the value is calculated since last measurement.
4. The server names on the WebSphere endpoint must be unique in the workspace of an instance.

Standalone client

The next picture shows the setup where one instance of PMIExplorer runs as standalone client:

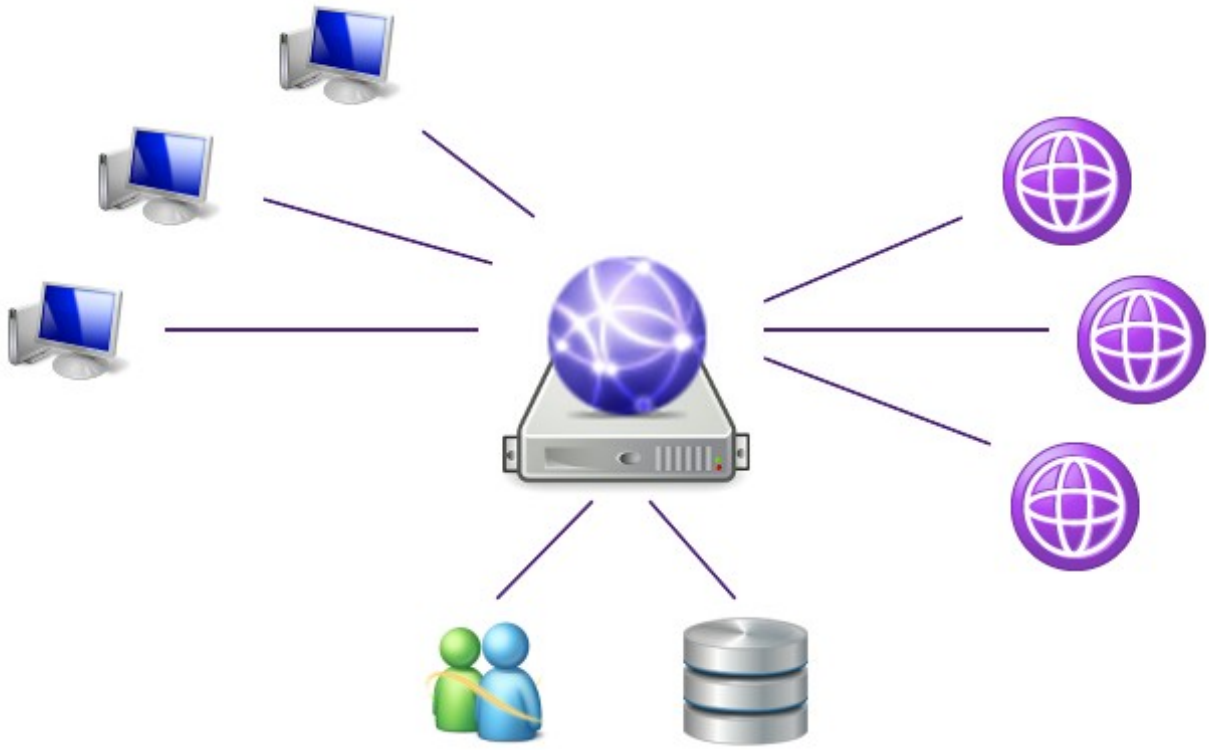


The client connects directly to the WebSphere environment(s) and records the values into the local database. This database might reside on a different (virtual) machine, but there is only one client which retrieves the information from that database directly.

It is possible for a standalone client to connect to a remote agent. The client can either connect to WebSphere to collect the performance data or connect to a remote agent to retrieve the data from there. It is not possible to connect to a WebSphere endpoint and the remote agent at the same time or connect to multiple remote agents.

Remote agent

The next picture shows the remote agent topology:



This setup consists of multiple instances of PMIExplorer. The client(s) connect to a central remote agent installation. This remote agent connects to WebSphere and continuously records information from the WebSphere environment(s). The recorded samples are stored in a relational database.

It's recommended to setup an IBM DB2, Oracle or Microsoft SQL server for this topology. Only the remote agent connects directly to this database and clients retrieve the data via service calls on the remote agent.

The remote agent can connect to a directory server (using LDAP) and enforce client authentication and protect the performance data and monitoring configuration from unauthorized access.

Installing PMIExplorer

Downloading the software

The PMIExplorer software is available from the website: www.pmiexplorer.com. You need to register yourself on the website before you can download the software. PMIExplorer will also provide updates via the website.

The download is available in a `zip` package.

Installing the software

PMIExplorer has chosen to distribute the software in a compressed archive which does not require administrator privileges to install on your system(s). You can extract this archive in your preferred location. The archive has the following contents:

File system structure:	Description:
`PMIExplorer`	The `<pmi_home>` directory
`config`	Contains the PMIExplorer config files
`data`	Internal database storage
`ext`	Location for external libraries
`keys`	License keys and certificates
`lib`	Third party libraries
`logs`	Log location for PMIExplorer
`profiles`	Default location for monitor profiles
`pmiexplorer.jar`	The executable jar file

WebSphere connector jars

In order to connect to a WebSphere environment you have to copy two WebSphere libraries into the `ext` folder of your PMIExplorer installation. These jars contain the AdminClient which is used to securely connect over SOAP and read the performance data from your WebSphere installation.

Library name	WebSphere location	PMIExplorer location
com.ibm.ws.admin.client-x.x.x.jar	`<was_home>`/runtimes	`<pmi_home>`/ext
com.ibm.ws.security.crypto.jar	`<was_home>`/plugins	`<pmi_home>`/ext

In the table above the `<was_home>` refers to the location where you have installed your WebSphere application server binaries; for example `/opt/IBM/WebSphere/AppServer`. The `<pmi_home>` refers to the location where you have installed the PMIExplorer; for example `/opt/PMIExplorer`.

Java runtime environment

Choosing your Java runtime

PMIExplorer requires a Java 7 runtime environment. The Java runtime is not shipped as part of the product due to licensing restrictions. PMIExplorer has been verified to run on the following Java virtual machines:

- Oracle Java 7 – http://java.oracle.com
- IBM Java 7 – http://www.ibm.com/developerworks/java/jdk

In order to connect to a WebSphere endpoint you must use the IBM Java 7 Runtime Environment, since the AdminClient connection that is used to connect to your WebSphere installation imposes this requirement.

NOTE:

IBM does not deliver the JDK for Windows environments as a separate installable product. In order to use the remote agent on the Windows platform PMIExplorer recommends installing the WebSphere AppClient on the Windows machine and use the JDK that is shipped together with the WebSphere AppClient. Using the AppClient ensures easy updates to the Java runtime if (security) fixes become available. It's also possible to obtain the Windows JDK from the IBM Development Package for Eclipse.

Installing the Java runtime

How to install Java depends on your Java vendor and platform. For windows machines you can download and install from an executable and Linux distributions often have rpm or deb packages.

It is possible to install multiple versions of Java on one machine. To run with a specific Java version you need to invoke PMIExplorer with that specific version i.e.:

```
/usr/java/jdk-x86_64-70/bin/java -jar pmiexplorer.jar
```

Please refer to the documentation from your Java vendor for your platform for detailed installation instructions.

Preparing the database

Choosing your DBMS

PMIExplorer supports several database management systems. Depending on your organizations requirements you can store the PMIExplorer data in your corporate database system. You can also choose to use the internal JavaDB / Apache Derby database.

PMIExplorer recommends using IBM DB2, Microsoft SQL Server or Oracle database for continues monitoring in a remote agent topology. In our experience the internal database cannot deliver the same performance as the other database systems.

PMIExplorer has been tested in an enterprise environment using an Oracle Enterprise database where it processed 200.000.000 samples a week. The database grew to 200 GB and contained over 3.400.000.000 samples.

When choosing your database system PMIExplorer considering the following questions:

1. For how long would you like to keep the data in your database? The current version of PMIExplorer cannot compress data over time yet.
2. Is it possible to collocate the data of PMIExplorer with the data of another application; possible with an application monitored by PMIExplorer?
3. What is your strategy for backup and recovery. Do you really need to recover the PMIExplorer data in case of disaster?
4. Do you want to PMIExplorer to modify the database objects or would you like to have manual control over the database object changes?

5. Would you like PMIExplorer to invoke database statistics tasks every midnight when performing its maintenance cycle?
6. Would you like to implement custom logic to support partitioning or other advanced database features?

JDBC drivers

A JDBC drivers is required to connect to the database. PMIExplorer automatically loads these drivers when they have been placed in the `<pmi_home>/ext` directory. Due to licensing restrictions PMIExplorer.com cannot redistribute these JDBC drivers. You are therefore required to obtain these drivers from your database administrator or database vendor.

The following libraries are commonly used for the supported database versions:

Database system	JDBC Libraries
IBM DB2 UDB	• db2jcc.jar • db2jcc4.jar • db2jcc_license_cu.jar
Oracle database	• ojdbc7.jar
Microsoft SQL Server	• sqljdbc4.jar

Creating tables manually

Some database administrators do not like that an application can modify the database objects and would prefer to review or alter the scripts to support partitioning for example. PMIExplorer has the ability to automatically create all required database artifacts, but can also generate a script file for review by a DBA before execution. When the property `pmi.jdbc.create` is set to `false` then database script file(s) are generated in the data directory when there are updates to the database due to added PMIExplorer functionality.

When changes to the database are pending then PMIExplorer cannot start as long as the changes have not been applied to the database.

Preparing the internal database

The internal database is configured by default and does not need much preparation. PMIExplorer works with reasonable defaults using an 8K block size and 32.768 page cache size. This means the internal database could potentially use 258 Mb of the heap.

During the midnight maintenance cycles PMIExplorer invokes the database routines to compress the database storage and run statistics on all tables.

You could consider moving the internal database to fast(er) storage by changing the database URL. Please refer to the JavaDB documentation for configuration details and options for performance tuning.

Preparing IBM DB2 UDB

In order to use an IBM DB2 database you need to adjust your configuration file accordingly. See the next chapter for detailed descriptions of the configuration properties. Additionally you need to setup your database with the steps below.

Create user

IBM DB2 relies on your operating system for user accounts. Please refer to you platform documentation on how to create a user account to be used in IBM DB2. Please note that IBM DB2 has some restrictions with regard to user names.

Create database

Invoke the following command in order to create the database. Please consult your database administrator or DB2 documentation for additional database creation parameters.

```
db2 create database <database name>;
```

Assign permissions

In order to access the database the user account must have the proper authorities. The account should at least be allowed to connect (`CONNECT`), but you could also grant the permissions to create tables (`CREATETAB`) or allow the user to administer the database (`DBADM`). Execute the following statement to assign the permissions.

```
db2 connect to <database name>;
db2 grant <permission> on database to <username>;
```

NOTE:

permission must be replaced with CONNECT for read privileges, CREATETAB for creating database artifacts or DBADM allow full access to the database.

Create schema

Invoke the following command in order to create the schema. Please consult your database administrator or DB2 documentation for additional schema creation parameters.

```
db2 create schema <schema> authorization <username>;
```

Preparing the Oracle Database

In order to use an Oracle database you need to adjust your configuration file accordingly. See the next chapter for detailed descriptions of the configuration properties. Additionally you need to configure your database with the steps below.

Create user

In order to connect to the database you require a user account. Execute the following statement to create an Oracle database user account.

```
create user <username> identified by <password>;
```

Assign permissions

In order to access the database the user must have the proper authorities. At least the user should be allowed to connect (`CONNECT, RESOURCE`), but you could also grant the permissions to create tables (`CREATE TABLE, CREATE SEQUENCE, CREATE INDEX`) or allow the user to administer the database (`DBA`). Execute the following statement to assign a permissions.

```
grant <permission> to <username>;
```

NOTE:

permission must be replaced with CONNECT, RESOURCE for read privileges, CREATE TABLE, CREATE SEQUENCE, CREATE INDEX for creating database artifacts or DBA allow full access to the database (not recommended).

Create schema

When a user is created in Oracle then a schema is also created for that user automatically. You can use this schema to store the recorded performance data. It is also possible to setup another schema for the data, but you need to set this schema as default schema for the database user that connects to the database.

Preparing Microsoft SQL Server

In order to use a Microsoft SQL Server database you need to adjust the configuration file accordingly. See the next chapter for detailed descriptions of the configuration properties. Additionally you need to setup your database with the steps below:

Step 1: Create database

Create a new database and set the options to your organizations defaults.

Step 2: Create login

Create a login to access the PMIExplorer data. It is possible to use Windows authentication or SQL Server authentication. PMIExplorer requires a username and password to access the database over JDBC. Set the database from step 1 as the default database.

Step 3: Create user

Create a user in the database which was created in step 1. Link the user to the login created in step 2. Set the permissions to `db_owner` or `db_datawriter`.

Step 4: Create schema

Create the schema for the PMIExplorer tables. Specify the user from step 3 as schema owner.

Step 5: Set the default schema for the user

Update the user created in step 3 and set the schema created in step 4 as the default schema.

Preparing security

LDAP connectivity

PMIExplorer can use an existing user repository which is accessible via LDAP to authenticate users accessing the remote agent. Users and groups in the directory server can be linked to the roles available in PMIExplorer when configuring the authorizations.

Security is only available on the remote agent. The standalone client is not accessible by others and runs on an operating systems which is presumed to have its own authentication and authorization mechanisms.

In order to setup LDAP security you need to configure the properties described in chapter LDAP Security on page 24

Authorizations

PMIExplorer provides three roles which can be linked to the users and groups in your directory. Using these roles you can deny all access, grant users read only access (`reader`), allow them to make changes to the endpoints, monitors and alarms (`editor`) or grant them full access (`administrator`).

Details about PMIExplorer security and available roles are described in more detail in chapter Security on page 72.

In order to grant permissions in PMIExplorer you might want to create additional groups in your directory.

Certificates

Access to your WebSphere environment requires an SSL connection. PMIExplorer can automatically accept the certificates provided by your WebSphere environment. By default the configuration property `pmi.ssl.automatic` is set to true to automatically accept SSL certificates. To deny access to certain WebSphere environment you can disable this configuration and provide the certificates for the allowed WebSphere environments in a JKS keystore. Also secure connections to LDAP are automatically accepted or require the certificate to be available in the JKS keystore.

When automatic certificate acceptation is disabled for a remote agent with security enabled you must prepare the keystore before starting up PMIExplorer, because authentication depends on certificate availability when LDAP SSL is configured.

Importing the license

Obtaining your license key

License keys for PMIExplorer are available on www.pmiexplorer.com. You need to create an account and configure the license online. Once the license fee has been payed the license key becomes available for download.

You can download your license key separate from the software. PMIExplorer recommends you first download and install the software before obtaining and loading the license key.

NOTE:

The license keys have a limited activation window of 2 or 3 days depending on type of license. It is recommended to first install the software and obtain your license key once the software is operational.

Loading the license key

How to load the license key depends on your topology. When using the standalone client you need to import the license key via the `manage license` dialog. When using the remote agent topology the license key is requested when you connect to the agent for the first time.

The features of PMIExplorer related to connecting to WebSphere endpoints will be unlocked once the software has been activated with the license key.

Please read the chapter `Managing licenses` on page `69` for more details about the licensing system and activation keys.

Configuration

Configuration files

PMIExplorer.conf

The PMIExplorer.conf file contains the configuration for PMIExplorer. The file is loaded when the application starts. Modifications made to the file while running PMIExplorer are not reloaded while the application is running.

PMIExplorer will look for a file named `PMIExplorer.conf` in the `config` directory by default, but it is possible to work with multiple configuration files. In order to override the default configuration file name it can be passed as a command line parameter:

```
java -jar pmiexplorer.jar config/myfile.conf
```

User interface

pmi.country

The country and language properties determine the how dates and numbers are formatted in the user interface. Currently the user interface is only available in English, but future versions of PMIExplorer might support more languages.

```
VALUE TYPE:      String
DEFAULT VALUE:   US
```

pmi.language

The country and language properties determine the how dates and numbers are formatted in the user interface. Currently the user interface is only available in English, but future version of PMIExplorer might support more languages.

```
VALUE TYPE:      String
DEFAULT VALUE:   en
```

pmi.chart.size

The chart size property determines the time span of the chart to plot when viewing real-time activity on the WebSphere environment(s). You can change the value depending on your needs which might depend on your screen resolution for example. By default PMIExplorer will show activity from the last 30 minutes.

```
VALUE TYPE:      Number (seconds)
DEFAULT VALUE:   1800
```

linux.tray.color

Java on Linux cannot render a transparent background for the system tray icon. With this property you can override the background color for the tray icon to match the color on your desktop. This value requires 6 digits (without #-sign) and defaults to black (000000) for the system tray icon if no value was provided.

```
VALUE TYPE:      Number (6 digits)
DEFAULT VALUE:   000000
```

PMI Collector

pmi.collector.interval

This property determines how often PMIExplorer reads PMI data from your WebSphere installation. PMIExplorer keeps an open AdminClient connection to the WebSphere environment and collects metrics every 10 seconds by default.

```
VALUE TYPE:      Number (seconds)
DEFAULT VALUE:   10
```

pmi.collector.reload

PMIExplorer periodically scans your WebSphere environment and updates the tree with available metrics if new metrics have become available. This happens when some functions are invoked for the first time or when a node in the cell is restarted. By default the scan is performed every 5 minutes. The reload functionality can be switched off by specifying the value 0.

```
VALUE TYPE:      Number (seconds)
DEFAULT VALUE:   300
```

pmi.collector.reconnect

PMIExplorer can reconnect to a WebSphere environment when it has lost the connection. Reconnecting is integrated in the reload functionality. During this periodical scan PMIExplorer also checks if the endpoints are still alive. If an endpoint is not alive and it did not fail with an authorization error then PMIExplorer attempts to reconnect to the WebSphere environment. Reconnecting is enabled by default and can be disabled by setting this property to false or disabling the reload functionality.

```
VALUE TYPE:      Boolean
DEFAULT VALUE:   true
```

pmi.collector.retention

The retention parameter determines how long PMIExplorer stores the data in its database. Every night at midnight a task is launched which prunes records older then the configured amount of days. Currently PMIExplorer does not have capabilities to compress data yet. This feature will be added to a future release. By default PMIExplorer saves the collected samples for 30 days. Pruning records can be switched off by specifying the value 0.

```
VALUE TYPE:      Number (days)
DEFAULT VALUE:   30
```

pmi.collector.cache

Once data has been collected from your WebSphere environment it will be cached for some time in the memory of PMIExplorer. This cached data is used for exporting real-time statistics and viewing historical details. The size of the cache should be at least 2 times the granularity of the export (see the export chapter for details). The PMIExplorer heap needs to be sized big enough to hold this data. Caching data can be switched off by specifying the value 0.

```
VALUE TYPE:      Number (seconds)
DEFAULT VALUE:   300
```

pmi.roots

PMIExplorer includes a set of “roots” which enable to detection of all default WebSphere Application Server metrics. This list of roots can be extended to include your custom PMI modules or “roots” from other WebSphere products. Adding `WBIStat.RootGroup` for example enables collecting metrics from WebSphere Process Server BPEL flows.

```
VALUE TYPE:      String (comma separated)
DEFAULT VALUE:
```

pmi.ssl.automatic

When connecting to your WebSphere environment the connection is secured with SSL certificates. In order to establish the connection PMIExplorer needs to trust that connection. You can either import all trusted certificates in the PMIExplorer keystore or trust all hosts the PMIExplorer connects to. When you trust all hosts then you can accept all certificates automatically by specifying the value `true`. Otherwise set this value to `false`. By default PMIExplorer accepts certificates automatically. Automatic certificate acceptation also applies to SSL secured connections to a directory server over LDAPS.

```
VALUE TYPE:      Boolean
DEFAULT VALUE:   true
```

pmi.ssl.keystore

The SSL keystore is required to store the trusted certificates. When you do not automatically accept SSL certificate automatically then you need to add these certificates to the keystore manually, or provide the path to another keystore. The only allowed type of keystore is JKS. By default PMIExplorer provides a keystore filled with certificates from some commonly used certificate authorities.

```
VALUE TYPE:      String
DEFAULT VALUE:   keys/certificates.jks
```

pmi.ssl.password

In order to open the keystore a password is required. PMIExplorer recommends you to change the password for the keystore when you have chosen not to automatically accept all SSL certificates when connecting.

```
VALUE TYPE:      String
DEFAULT VALUE:   changeit
```

Database connectivity

pmi.jdbc.db

Determines the type of database system. PMIExplorer supports the Internal Java database, IBM DB2 UDB, Microsoft SQL Server and Oracle Database. By default PMIExplorer uses the Internal database, but PMIExplorer recommends using DB2, SQL Server or Oracle for continues monitoring of your WebSphere environments.

```
VALUE TYPE:      String (Internal, DB2, MSSQL or Oracle)
DEFAULT VALUE:   Internal
```

pmi.jdbc.schema

This property specifies the database schema which contains the data for the PMIExplorer. For the internal database this property is disregarded and the schema name defaults to PMI.

```
VALUE TYPE:      String
DEFAULT VALUE:
```

pmi.jdbc.create

PMIExplorer can generate the database objects when they do not exist. Some database administrators prefer not to automatically create the database objects by an application and therefore PMIExplorer has the option not to create database objects automatically. When this property is set to `true` PMIExplorer will create (or update) database objects when necessary. When this property is set to `false` PMIExplorer will create a database script file in the data directory which contains the statements to manipulate the database objects.

```
VALUE TYPE:      Boolean
DEFAULT VALUE:   true
```

pmi.jdbc.driver

This property specifies the driver for the database connection. PMIExplorer does not require drivers with XA capabilities, but the driver must be compatible with your database system. The driver jar file will be loaded by PMIExplorer when it is placed in the `ext` directory. The following drivers are commonly used:

- Internal: org.apache.derby.jdbc.EmbeddedDriver
- DB2: com.ibm.db2.jcc.DB2Driver
- MSSQL: com.microsoft.sqlserver.jdbc.SQLServerDriver
- Oracle: oracle.jdbc.driver.OracleDriver

```
VALUE TYPE:      String
DEFAULT VALUE:   org.apache.derby.jdbc.EmbeddedDriver
```

pmi.jdbc.username

PMIExplorer requires a username to connect to the database. When connecting to the Internal database this property is disregarded

```
VALUE TYPE:      String
DEFAULT VALUE:
```

pmi.jdbc.password

PMIExplorer requires a password to connect to the database. When connecting to the Internal database this property is disregarded

```
VALUE TYPE:      String
DEFAULT VALUE:
```

pmi.jdbc.url

The JDBC URL specifies where the database is located and how to access it. JDBC URLs have a vendor specific formatting. The following samples could be used:

- Internal: jdbc:derby:data/pmi;create=true
- DB2: jdbc:db2://server.company.com:50001/pmi
- MSSQL: jdbc:sqlserver://server.company.com:1433;databaseName=pmi
- Oracle: jdbc:oracle:thin:@server.company.com:1521/pmi

```
VALUE TYPE:      String
DEFAULT VALUE:   jdbc:derby:data/pmi;create=true
```

pmi.jdbc.pool.minimum

Depending on the usage of PMIExplorer you might require more database connections. This parameter controls the minimum amount of connections in the connection pool. Depending on the amount of endpoints, workspaces, users, alarms and export configuration you might want change the lower bound of the connection pool.

```
VALUE TYPE:      Number
DEFAULT VALUE:   10
```

pmi.jdbc.pool.maximum

Depending on the usage of PMIExplorer you might require more database connections. This parameter controls the maximum amount of connections in the connection pool. Depending on the amount of endpoints, workspaces, users, alarms and export configuration you might need to change the upper bound of the connection pool.

```
VALUE TYPE:      Number
DEFAULT VALUE:   50
```

pmi.jdbc.pool.statementCache

PMIExplorer caches its prepared statements for performance reasons. PMIExplorer could benefit from an increased statement cache size, but there is a trade off with memory consumption. This parameter gives control over this behavior.

```
VALUE TYPE:      Number
DEFAULT VALUE:   50
```

pmi.jdbc.parallelDataRetrievers

When retrieving historical information from the database PMIExplorer can utilize multiple connections to build the timeline faster. This property controls how many connections retrieve the data parallel. The minimum time span for one retriever is set to 1 hour, which means that retrieving one hour of historical information is performed by one connection, even if this property is set to a higher value. When viewing performance data from the last 24 hours PMIExplorer uses 16 connections to retrieve even chunks of data.

```
VALUE TYPE:      Number
DEFAULT VALUE:   16
```

pmi.jdbc.parallelDataWriters

Storing the collected data is performed asynchronously from retrieving the data. In case of a temporary database outage PMIExplorer will continue collecting data and is not affected by the unavailability of the database. Between retrieving and persisting the data is stored in a buffer. This property defines how many data writers read from the buffer to store the samples in the database. You might want to increase this value if you have a huge number of endpoints or servers in your workspace(s).

```
VALUE TYPE:      Number
DEFAULT VALUE:   2
```

pmi.jdbc.dataBufferSize

Storing the collected is performed asynchronously from retrieving the data. In case of a temporary database outage PMIExplorer will continue collecting data and is not affected by the unavailability of the database. Between retrieving and saving, the data is stored in a buffer. All measurements for one server are stored in one entry in the buffer. When the buffer is full these measurements are discarded and not persisted to the database when the database becomes available again. This property defines the size of the buffer.

```
VALUE TYPE:      Number
DEFAULT VALUE:   1000
```

pmi.jdbc.runNightlyStatistics

PMIExplorer can invoke database functions to gather statistics on all supported databases. This property controls if PMIExplorer should collect database statistics every night, after it performs other maintenance tasks. When using the internal database this property is ignored and statistics are gathered automatically at midnight. Please consult your DBA about before enabling this feature since there might be other policies in effect to gather statistics.

VALUE TYPE: Boolean

DEFAULT VALUE: false

pmi.jdbc.pruneProcedure

PMIExplorer supports invocation of a stored procedure on the database to perform the housekeeping on the database. This procedure takes one parameter containing the timestamp. The procedure should prune all records before this timestamp and can perform additional tasks like dropping partitions, gathering statistics or rebuilding indexes. The procedure needs to be available in the same schema as the data specified by `pmi.jdbc.schema`.

VALUE TYPE: String

DEFAULT VALUE:

LDAP Security

Using LDAP security you can restrict access to your monitoring data and configuration on the remote agent. When security is enabled PMIExplorer provides three roles which are available on the remote agent. See the security chapter for more details on setting up security for the remote agent. The related configuration properties are described below.

PMIExplorer does not attempt to modify information in your directory. It solely uses the directory for authentication and authorization purposes.

pmi.ldap.host

This property specifies the host of your directory server. You can specify a single LDAP server or provide a hostname or IP address to connect to a load balanced LDAP service. By default this field is empty

VALUE TYPE: String

DEFAULT VALUE:

pmi.ldap.port

This property specifies the port of your directory server. Usually the port number is `389` for unsecured connections and `636` for SSL protected connections. By default this property is not filled.

VALUE TYPE: Number

DEFAULT VALUE:

pmi.ldap.ssl

Access to your LDAP server could require the usage of SSL connection. This property

specifies if SSL is used. PMIExplorer recommends using SSL for LDAP connections since sensitive information like user names and passwords are send to the directory server during authentication.

VALUE TYPE: Boolean

DEFAULT VALUE: true

pmi.ldap.baseDN

This property specifies the base distinguished name of your directory. This is the root element which is used to query your directory for information about users, groups and group membership. Please refer to the documentation of your LDAP server for syntax details and your LDAP administrator for your setup. The baseDN could resemble something like: `o=pmiexplorer.com,c=nl`

VALUE TYPE: String

DEFAULT VALUE:

pmi.ldap.bindUserDN

In order to search your directory you might need to authenticate first. For authentication you need to specify the fully qualified distinguished name of the bind account, for example `cn=pmibind,ou=accounts,o=pmiexplorer.com,c=nl`. PMIExplorer uses this bind account to load user details and determine group memberships. When the `bindUserDN` and `bindPassword` are not specified PMIExplorer attempts to bind anonymously.

VALUE TYPE: String

DEFAULT VALUE:

pmi.ldap.bindPassword

If you are required to authenticate to your directory you must provide the password. If the `bindUserDN` and `bindPassword` properties are not specified an anonymous bind will be attempted.

VALUE TYPE: String

DEFAULT VALUE:

pmi.ldap.userFilter

The user filter defines how to uniquely identify users in your directory. A search query is performed to find the distinguished name for users who need to authentication and authorization. If the search has multiple entries than access is denied for that user. The user filter could look like `(&(uid=%v)(objectclass=inetOrgPerson))`. Please refer to your LDAP

documentation or administrator for more details on the user filter.

VALUE TYPE: String

DEFAULT VALUE:

pmi.ldap.groupFilter

The group filter is used to find groups in your directory. A search query is performed to locate the group in the directory and find the members of the group. This filter could look like `(&(cn=%v)(|(objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))`, where all occurrences of `%v` are replace with the group name. Please refer to your LDAP documentation or administrator for more details on the group filter.

VALUE TYPE: String

DEFAULT VALUE:

pmi.ldap.groupMembers

This property defines the group membership attribute(s). The members of a group could be users or other (nested) groups. PMIExplorer recursively checks the group members until it cannot find entities with with a member attribute anymore. A user is considered a group member when its distinguished name equals the value of the group member attribute. It is possible to specify multiple attributes for this property separated with commas. The common attributes for group membership are `member` and `memberUid`.

VALUE TYPE: String (comma separated)

DEFAULT VALUE:

pmi.administrator

This property defines the login name of the PMIExplorer primary administrator. The account configured here becomes the primary administrator which is allowed to setup the initial security configuration and import the license key. See the security chapter for more details about administrator permissions within PMIExplorer.

VALUE TYPE: String

DEFAULT VALUE:

Extension points

pmi.data.exporters

With this property you can specify your custom data exporter classes. Custom data exporter enable integration with incident management systems within your organization.

In order to load the data exporters you need to specify the fully qualified name of the class. Multiple exporters can be separated using a comma.

```
VALUE TYPE:      String (comma separated)
DEFAULT VALUE:
```

pmi.alarm.executors

With this property you can specify your custom alarm executors. Custom alarm executors enable integration with other systems in your organization. In order to load the alarm executor you need to specify the fully qualified name of the class. Multiple executors are separated with a comma.

```
VALUE TYPE:      String (comma separated)
DEFAULT VALUE:
```

Remote agent

pmi.agent.enabled

For an instance of PMIExplorer to act as remote agent you need to enable the agent service. The agent uses IIOP (Internet Inter ORB Protocol) for clients to communicate with the agents service. PMIExplorer automatically starts a IIOP name service when launching the remote agent. More details are available in the remote agent communications chapter. The PMIExplorer agent runs headless / without graphical user interface.

```
VALUE TYPE:      Boolean
DEFAULT VALUE:   false
```

pmi.agent.host

Defines the hostname or IP address of the remote agent. In case of a multi-homed system you can specify the IP address to listen on. When left empty PMIExplorer cannot start the remote agent service.

When this property is specified in a client installation the value used in the connection dialog to allow users to connect to the remote agent quickly.

```
VALUE TYPE:      String
DEFAULT VALUE:
```

pmi.agent.port

Defines the port of the remote agent. Clients will initially connect to this port to establish

communication with the remote agent. Another port for bi-directional communication is negotiated when connecting to the remote agent. When left empty PMIExplorer cannot start the remote agent service.

When this property is specified in a client installation the value used in the connection dialog to allow users to connect to the remote agent quickly.

VALUE TYPE: Number

DEFAULT VALUE:

pmi.agent.workspaces

This property defines the names of the workspaces that are available on the remote agent. Workspaces for the remote agent can only be defined in the configuration file and cannot be created on demand. When this property is set for a client installation this information is used in the connection dialog. You can define multiple workspaces with a comma separated list.

VALUE TYPE: String (comma separated)

DEFAULT VALUE:

pmi.agent.persistent

This property defines if the remote agent persists the monitoring configuration. When set to `true` the endpoint connection details (including username and password), monitoring profiles and active exporters are stored in the database. This information is read from the database when the agent is restarted. The agent then automatically re-connects to the endpoints, applies the monitoring profile and resumes the export. The username and password are stored in encrypted form using an AES cipher. Set this property to `false` to avoid saving this information or if automatic restart of the monitoring is not desired.

VALUE TYPE: Boolean

DEFAULT VALUE: true

Notifications

pmi.smtp.host

This property specifies the hostname of your email server. You can specify the SMTP host by IP address or any other name which identifies your email server. By default this value will be empty.

VALUE TYPE: String

DEFAULT VALUE:

pmi.smtp.port

This property specifies the port of your SMTP server. When using unsecured SMTP the number of this port is usually 25. Secure SMTP often uses port 587. By default this value will be empty

VALUE TYPE: Number

DEFAULT VALUE:

pmi.smtp.ssl

This property determines if the connection to the email server is secured. When SMTP security is enabled PMIExplorer attempt to use SSLv3 and TLSv1 protocols with starttls to setup a secure connection to your SMTP server. When this property is disabled the communication between PMIExplorer and the SMTP server is not encrypted.

VALUE TYPE: Boolean

DEFAULT VALUE: true

pmi.smtp.username

This property specifies the username for the SMTP connection. If your e-mail server requires authentication this value must be provided. For access to e-mail servers without authentication this property must be left empty. PMIExplorer recommends providing credentials only over secured or trusted connections.

VALUE TYPE: String

DEFAULT VALUE:

pmi.smtp.password

This property specifies the password for the SMTP connection. If your e-mail server requires authentication this value must be provided. For access to e-mail servers without authentication this property must be left empty. PMIExplorer recommends providing credentials only over secured or trusted connections.

VALUE TYPE: String

DEFAULT VALUE:

pmi.smtp.sender

This property defines the sender address. Emails send from PMIExplorer will originate from this address when sending alerts.

VALUE TYPE: String

DEFAULT VALUE:

pmi.smtp.receiver

This property defines the receiver address. Emails will be delivered to this address when sending notifications from PMIExplorer.

VALUE TYPE: String

DEFAULT VALUE:

Logging

pmi.log.level

This property defines how much information is written to the PMIExplorer log file. Available log levels are `WARN`, `INFO`, `DEBUG` and `TRACE`. Increasing the log level has an impact on performance and it's not recommended to run log levels higher than `INFO`. For support purposes PMIExplorer.com might ask you to increase the default log level.

VALUE TYPE: Number

DEFAULT VALUE: INFO

Running PMIExplorer

Starting the client

The PMIExplorer client can be started by double clicking the pmiexplorer.jar. That will launch the client with your operating system settings for Java. When the client needs to connect to a WebSphere environment these default might not meet the requirement to use an IBM Java 7 runtime.

It is also possible to launch the PMIExplorer from the command line with the following command. This command ensures usage of an IBM Java 7 runtime and sets the heapsize.

```
/usr/java/jdk-x86_64-70/bin/javaw -Xmx512M -jar pmiexplorer.jar
```

To launch PMIExplorer with different configuration file use the following command:

```
javaw -jar pmiexplorer.jar config/filename.conf
```

NOTE:

When launching PMIExplorer with the `javaw` *command instead of* java *the console or dos box is not shown. PMIExplorer writes all messages to a log file which is located in the logs directory. The PMIExplorer client should run fine with a 256M or 512M heapsize. The amount of heap needed depends mostly on the amount of metrics monitored. Rendering the details in the user interface is a bit memory hungry. Monitoring 750 metrics / 3000 statistics requires about 512 Mb.*

Starting the remote agent

Start the remote agent from the command line in the background with the following command or implement a service wrapper:

```
nohup /usr/java/jdk-x86_64-70/bin/java -Xmx512M -jar pmiexplorer.jar &
```

or launch with a different configuration file

```
nohup java -Xmx512M -jar pmiexplorer.jar config/filename.conf &
```

NOTE:

When launching PMIExplorer with the `java` *command together with* `nohup` *all output is written to the nohup.out file. PMIExplorer writes all messages to a log file which is located in the logs directory. The remote agent should run fine with a 512M or 1024M heap. The amount of heap needed depends mostly on the number of endpoints and the amount of connecting clients. Monitoring 750 metrics / 3000 statistics on 4 endpoints with 10 active (connected) clients requires about 1024 Mb.*

NOTE:

The remote agent remembers the monitoring state and will reload endpoints, monitoring profiles and exports when started up again. This behavior is controller with the pmi.agent.persistent property. The standalone client does not remember the active monitoring state.

Stopping the remote agent

To stop the remote agent invoke the following command. This shutdown command might be incorporated in an init script or Java service wrapper.

```
java -jar pmiexplorer.jar -shutdown
```

or initiate shutdown with a different configuration file

```
java -jar pmiexplorer.jar -shutdown config/filename.conf
```

The shutdown command connects to the running agent and initiates the shutdown sequence. The agent finishes the current recording and flushes the parallel writer buffer to the database before shutting down in order to ensure that no data is lost during the shutdown.

NOTE:

You can only shut down PMIExplorer from the host where is runs. The shutdown sequence validates the hostnames before halting its services.

Running as a service

Since PMIExplorer is based on Java there are many ways to launch the program as a service. Out-of-the box PMIExplorer does not provide a service implementation, but there are several solutions to run Java programs as a service like creating your own init scripts for UNIX / Linux or using one of these frameworks:

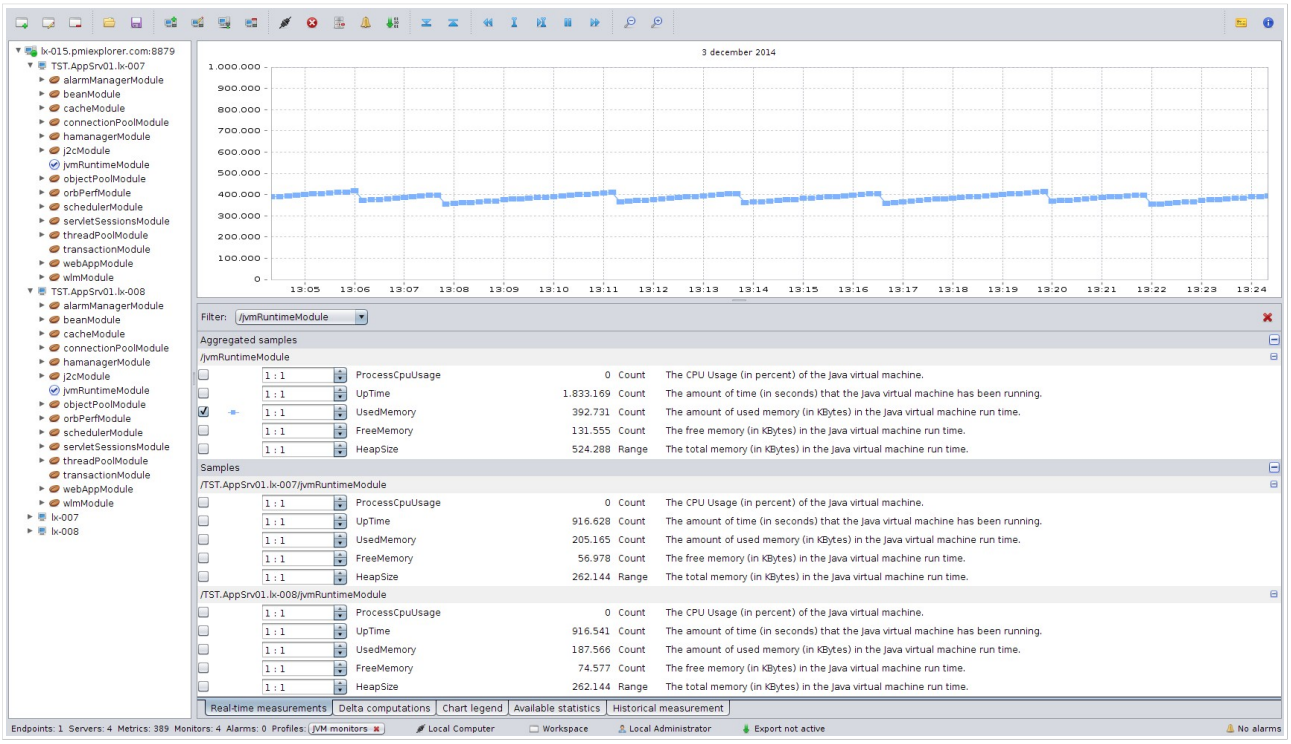
- Apache commons daemon
- Java Service Wrapper
- YAJSW – Yet Another Java Service wrapper

User interface

Layout

The PMIExplorer user interface is designed to resemble the interface in the admin console, but there are quite some differences because PMIExplorer provides a richer user interface.

The following image shows the user interface when monitoring the JVM runtime metric:


lx-015.pmiexplorer.com:8879
TST.AppSrv01.lx-007
alarmManagerModule
beanModule
cacheModule
connectionPoolModule
hamanagerModule
j2cModule
jvmRuntimeModule
objectPoolModule
orbPerfModule
schedulerModule
servletSessionsModule
threadPoolModule
transactionModule
webAppModule
wlmModule
TST.AppSrv01.lx-008
lx-007
lx-008
3 december 2014
Filter: /jvmRuntimeModule
Aggregated samples
/jvmRuntimeModule
ProcessCpuUsage 0 Count The CPU Usage (in percent) of the Java virtual machine.
UpTime 1.833.169 Count The amount of time (in seconds) that the Java virtual machine has been running.
UsedMemory 392.731 Count The amount of used memory (in KBytes) in the Java virtual machine run time.
FreeMemory 131.555 Count The free memory (in KBytes) in the Java virtual machine run time.
HeapSize 524.288 Range The total memory (in KBytes) in the Java virtual machine run time.
Samples
/TST.AppSrv01.lx-007/jvmRuntimeModule
ProcessCpuUsage 0 Count The CPU Usage (in percent) of the Java virtual machine.
UpTime 916.628 Count The amount of time (in seconds) that the Java virtual machine has been running.
UsedMemory 205.165 Count The amount of used memory (in KBytes) in the Java virtual machine run time.
FreeMemory 56.978 Count The free memory (in KBytes) in the Java virtual machine run time.
HeapSize 262.144 Range The total memory (in KBytes) in the Java virtual machine run time.
/TST.AppSrv01.lx-008/jvmRuntimeModule
ProcessCpuUsage 0 Count The CPU Usage (in percent) of the Java virtual machine.
UpTime 916.541 Count The amount of time (in seconds) that the Java virtual machine has been running.
UsedMemory 187.566 Count The amount of used memory (in KBytes) in the Java virtual machine run time.
FreeMemory 74.577 Count The free memory (in KBytes) in the Java virtual machine run time.
HeapSize 262.144 Range The total memory (in KBytes) in the Java virtual machine run time.
Real-time measurements
Delta computations
Chart legend
Available statistics
Historical measurement
Endpoints: 1 Servers: 4 Metrics: 389 Monitors: 4 Alarms: 0 Profiles: JVM monitors
Local Computer
Workspace
Local Administrator
Export not active
No alarms

The interface is divided into the following sections

Button bar

The button bar provides the main controls for the application. On the bar you can find the buttons to manage workspace, connect to the endpoint, navigate the chart and other features.

Tree panel

In the tree panel contains a hierarchical view of metrics that have been discovered on the endpoints. The tree contains at least three levels:

Component in the tree	Description
Endpoint	Connection to the WebSphere SOAP port
Server	Servers available in WebSphere environment
Metric	Available metrics; contains multiple levels

Within the tree panel you can load multiple endpoints in a workspace, by using the `Add endpoint` button on the button bar. Each endpoint contains one or more servers. The servers contain the metrics which can be selected for recording.

Chart panel

The chart panel plots the chart based on selections in the details panel. By default the chart shows the real-time activity and is continuously updated with the latest performance data from your WebSphere environment.

Using the chart buttons on the button bar you can control the chart. Possible actions include zooming in or out, moving back and forward in time and adjusting the scales and granularity.

You can click on the data points in the chart. That will retrieve all recorded data from that moment. You can also select a smaller timespan of the chart with your mouse which zooms in on that timespan.

Detail panels

The detail panels display the details of the recorded metrics. The `real-time measurements` and `delta computations` tabs are continuously updated with the latest recorded data. The `chart legend` shows the legend for the statistics currently charted and the `available statistics` tab gives an alternative way of locating the statistics. The information in the `historical details` tab is filled when a data point in the chart is clicked. At that moment all data recorded from that moment is loaded from cache or the database.

The grid on these panels has selection boxes which enable plotting a statistic on the chart panel. Also scaling options are provided to view statistics with different scales in one chart.

Status bar

The status bar contains information on the number of endpoints, servers and metrics that are currently monitored. Also the active monitoring profile is displayed and details on the current workspace are provided. The status bar also lists connection details for the remote agent (or `Local Computer`) and shows how many alarms have been fired.

Workspaces

The workspace concept

The workspace is an important concept within PMIExplorer. Most functions in PMIExplorer are bound to a workspace like monitoring profiles, alerting, endpoints, statistic aggregation and exports. When you use the standalone client you can create additional workspace on demand. Workspaces on the remote agent need to be predefined in the configuration file.

Within a workspace you can group WebSphere servers with a similar function together. That could be include one or more standalone servers, which serve a the same application, or a deployment manager containing which could contain one or more clusters. It is also possible to group servers from multiple cell topologies.

PMIExplorer automatically aggregates metrics which have the same path. When a workspace contains multiple servers with a `/threadpoolModule/WebContainer` metric then statistics for these metrics are also combined into one new aggregated statistic which provides the overall statistics for that metric for the workspace.

When you are connected to a workspace on a remote agent then you share this workspace with the other users. Changes made by other users in the workspace are propagated to all connected clients. When you start monitoring a metric for example then an event is relayed to other clients which also load the details for that metric. All the connected clients then start receiving updates for that metric.

To prevent that all user can make modifications to the workspace you can apply security roles to the workspace. There are roles available which entitle a user to view the data only or grant them permissions to make changes.

On the remote agent the workspaces need to be predefined with the `pmi.agent.workspaces` configuration parameter. This same parameter can also be used in the client configuration file where it's used to populate the workspaces dropdown when connecting to the remote agent.

Managing workspaces

PMIExplorer starts with an empty workspace and allows to create, close and rename workspaces on demand using the controls available in the button bar.

Opening a workspace

Use the button open new workspace to open a new workspace. When you click the button a new window is opened. In this window you can start a new monitoring session. When connected to the remote agent an additional pop-up will show, which asks you to select the name of one of the workspaces which are predefined on the remote agent. It is possible to open multiple workspaces with the same name on the remote agent.

Editing a workspace

Use the button rename workspace or switch workspace to control the workspace. The option to rename the workspace is only available in the standalone mode. This option customizes the name of the workspace. When connected to the remote agent you can use this button to switch between workspaces on the remote agent.

Closing a workspace

Use the button close this workspace to close a workspace. In the standalone client mode this will also stop the monitoring the endpoints in that workspace. When connected to the remote agent the monitoring continues on the remote agent. When you close the last workspace you are asked to confirm exiting the application. If you want to continue receiving alerts, but do not want to view the PMIExplorer window(s) you can also minimize to system tray.

PMI metrics tree

Endpoints, servers and metrics

The PMI metrics tree is populated when you connect to an endpoint. This endpoint could be a standalone WebSphere server or a deployment manager, which accessed via the SOAP port. When connected to the endpoint PMIExplorer starts a scanning for available

servers on that endpoint. In case of a standalone installation only one server is discovered and on distributed environments all servers and nodeagents known to the dmgr are detected. For each server the available metrics are retrieved and loaded in the tree.

The endpoint can have four connection states, which are reflected in the tree:

- The endpoint is disconnected
- The endpoint is connected to a WebSphere server or deployment manager
- The connection to the endpoint has failed
- The endpoint is paused

PMIExplorer reloads the servers and metrics every 5 minutes by default and updates the PMI metrics tree accordingly when new servers or metrics become available or unavailable. The reload interval can be configured with the property `pmi.collector.reload`. It is also possible to disable reloading with the same property by setting the value to zero. A button is available on the button bar to manually reload the tree.

PMIExplorer supports loading multiple endpoints in one workspace, as long as the server- and endpoint names are unique.

Managing the endpoints

Several buttons are available on the button bar to manage the endpoints in the tree:

Add endpoint

Use the button `add endpoint` to add an endpoint to the workspace. When you click on the button a dialog opens where you can specify the connection details of the endpoint. The name of the endpoint must be unique within and across all workspaces. It is possible to add the endpoint to the workspace without connecting, which is useful when you like to view back at previously recorded data.

Edit endpoint

Use the button `edit endpoint` to modify the state of the endpoint. When you click on the button a dialog opens which displays the current state of the endpoint. Depending on the state of the endpoint you can either pause the endpoint or reconnect. You might want to disconnect when you know your WebSphere endpoint goes down for maintenance.

Scan servers and metrics

Use the button `Scan servers and metrics` to manually reload the available servers and metrics on the connected endpoints. When you click on the button a background tasks starts which updates the PMI metrics tree when new servers or metrics are detected or a server or metrics is no longer available.

Remove endpoint

Use the button `Remove endpoint` to remove an endpoint from a workspace. When you click on this button a dialog opens which let you choose between only disconnecting or removing the endpoint from the workspace. When disconnected you do not

retrieve data anymore, but it still possible to view back data from that endpoint via the available statistics view.

Monitoring a metric

Starting monitoring is initiated from the PMI metrics tree. When you click on a node in the tree (which is not an endpoint or server) then the metric is registered with the collector component. The collector starts collecting the performance data for the metric and broadcasts events which update the data in the details panel. You can stop monitoring by clicking the same node in the tree again. The metrics in the tree can have three possible states:

- The metric is available for monitoring
- The metric is currently monitored
- The metric monitored as part of a monitoring profile

When the white icon is shown for a metric then it is not possible to stop monitoring by clicking on the node in the tree. You first need to disable the profile before you can stop monitoring these metrics. See chapter `Monitoring profiles` for more details about monitoring profiles.

Charting

Real-time charting

In order to start real-time charting you need to monitor one or more metrics. Expand the endpoints, servers and metrics in the tree panel until you reach the metric which you like to monitor. When you click on that metric the data collector starts taking measurements for that metric and the details are show in the details panels. The details panel contains a column with checkboxes which control which statistics are plotted in the chart area. When you check the boxes that will start plotting data on the chart in real-time.

It is also possible to plot historical data on the chart. Historical data is the data that does not fit in the chart anymore and could be 1 hour old or several days, weeks or months. Use the navigation buttons on the button bar to modify the timespan of the chart.

Chart navigation

In order to navigate the chart there are several buttons available on the button bar:

Pausing the chart

Use the button `Pause chart updates` to suspend updating the chart when new recordings come in. This will also stop the chart from moving along with the timeline. When you click on the button this button will toggle to a play button. Pausing the chart does not stop monitoring and the details panels will still be updated with the latest recorded data.

Playing the chart

Use the button `Play real-tine updates` to continue plotting the recordings on the chart. When you click the button that will start playing real-time updates and will therefore jump to the current timestamp. During the real-time updates the vertical scale is automatically adjusted. The speed of the chart is defined by the `pmi.collector.interval` configuration parameter. The default time span can be controlled with the `pmi.chart.size` parameter

Stepping back in time

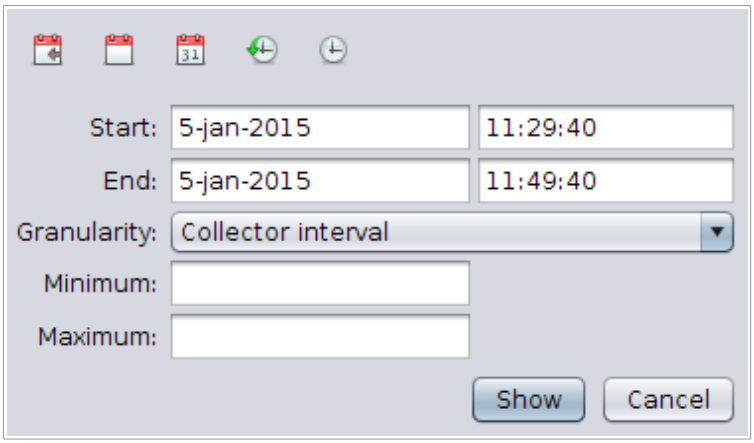
Use the button `Previous interval` to step back in time. Each time you click on the button you will go one step back in time. The size of the step is determined by the current time span of the chart which can be set by the `pmi.chart.size` parameter, the chart configuration dialog or by selecting an area in the chart with your mouse.

Stepping forward in time

Use the button `Next interval` to step forward in time. Each time you click on the button you will go one step forward in time. The size of the step is determined by the current time span of the chart which can be set by the `pmi.chart.size` parameter, the chart configuration dialog or by selecting an area in the chart with your mouse.

Customizing chart display

Use the button `Configure chart` to specify the charted area. When you click on the button that will open the chart configuration dialog. In the dialog you can enter the parameters which define the timespan, granularity (density of the data points) and the upper- and lower bounds of the chart.



Start: Defines the startpoint on the X-axis of the chart

End: Defines the endpoint on the X-axis of the chart

Granularity: Defines the density of the chart

Minimum: Defines the minimum value on the Y-axis of the chart

Maximum: Defines the maximum value on the Y-axis of the chart

The configure chart dialog comes with some presets when enable you to easily set some frequently used values. The available presets are (left to right on the button bar):

Yesterday:	Sets start- and endpoint to view yesterdays recordings
Today:	Sets start- and endpoint to view today's recordings
Round to day:	Rounds down the startpoint and round up the endpoint to whole days
Last hour:	Sets the start- and endpoint to view last hours recordings
Round to hours:	Rounds down the startpoint and rounds up the endpoint to whole hours

Customizing real-time display

Use the Configure real-time button to adjust the boundaries in during real-time charting. Normally PMIExplorer automatically calculates the minimum and maximum values for the Y-axis based on the values in the chart. You can override this behavior and specify the desired values. When you click on the button a pop-up dialog is opened which allows you to adjust the lower (minimum) and upper (maximum) boundaries. Leave one of the fields empty to keep computing that value automatically.

Mode:	Switches between manual and automatic boundaries
Minimum:	Defines the minimum value on the Y-axis of the chart
Maximum	Defines the maximum value on the Y-axis of the chart

Zooming in the chart

On the button bar there are button available to zoom in and out on the chart. Besides using the button you can also zoom in on a part of the chart by selecting that area with your mouse.

Zooming in

Use the button Zoom in to zoom in on the chart. Zooming in on the chart decreases the timespan of the chart and recalculates the minimum and maximum values for the Y-axis, unless a value was specified in the chart configuration. When zooming in on the chart the timespan is halved.

Zooming out

Use the button zoom out to zoom out on the chart. Zooming out on the chart increases the timespan of the chart and recalculates the minimum and maximum values for the Y-axis, unless a value was specified in the chart configuration. When zooming out on the chart the timespan is doubled.

Chart granularity

The chart granularity defines the amount of data points on the chart. By default the chart is displayed with the collector interval granularity, which defaults to 10 seconds.

When changing the granularity PMIExplorer adjusts the number of data points on the chart correspondingly. The following granularity presets are available:

- Collector interval (10 seconds default)
- 1 minute
- 2 minutes
- 5 minutes
- 15 minutes
- 1 hour
- 3 hours
- 6 hours
- 12 hours
- 1 day
- 1 week

Coarsening the granularity might be useful when viewing a larger timespan where you might not be interested in all the details. PMIExplorer computes the sum or average depending on the type of statistic. Refining the granularity is useful when you want to see more details. PMIExplorer cannot refine to a level beyond the collector interval, since that data is not recorded.

Coarsen granularity

Use the button `Coarsen granularity` to adjust the granularity of the chart. Coarsening the chart reduces the amount of data point plotted on the chart. When clicking the button the chart is recomputed based on the next available preset value. PMIExplorer re-calculates the values in the chart based on the type of statistic. For time- and range statistics the average of the data points is calculated and for count statistics the sum of the data points is calculated.

Refine granularity

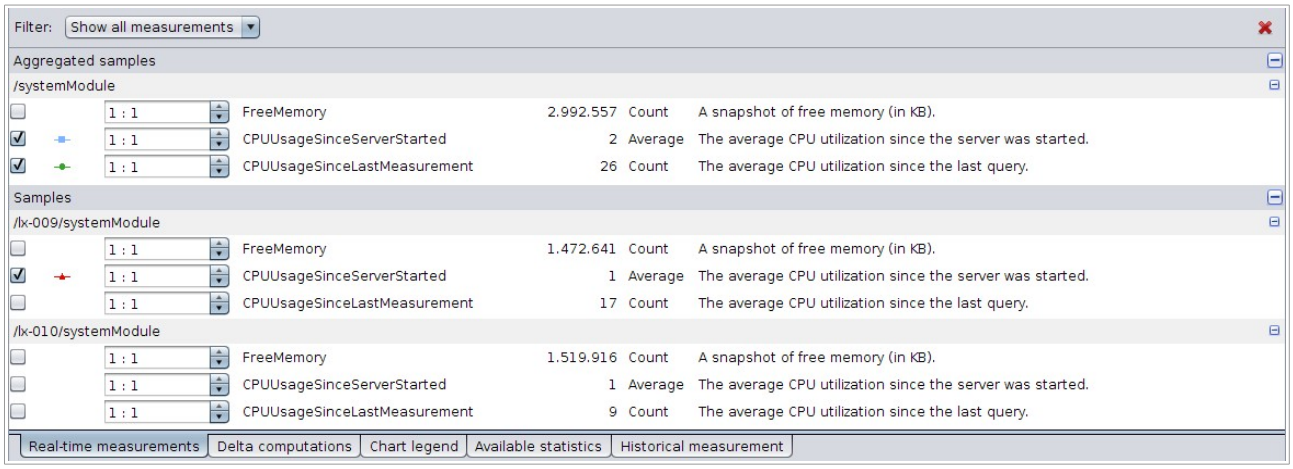
Use the button `Refine granularity` to adjust the granularity of the chart. Refining the chart increases the amount of data point plotted on the chart. When clicking the button the chart is recomputed based on the previous available preset value. PMIExplorer re-calculates the values for the chart based on the type of statistic. For time- and range statistics the average of the data points is calculated and for count statistics the sum of the data points is calculated.

Detail panels

PMIExplorer provides several views on the recorded data in the `detail panels`. Each panel interacts with the charted area where it can trigger an action on the chart or responds to an action from the chart.

Real-time measurements

The real-time measurements tab contains the details for the recorded data. The data on this tab is the data was recorded from WebSphere directly. The details are updated continuously when new performance data samples are read from WebSphere by the collector. The following image shows the recording of CPU usage from two nodes.


Filter:
Show all measurements
Aggregated samples
/systemModule
1 : 1 FreeMemory 2.992.557 Count A snapshot of free memory (in KB).
1 : 1 CPUUsageSinceServerStarted 2 Average The average CPU utilization since the server was started.
1 : 1 CPUUsageSinceLastMeasurement 26 Count The average CPU utilization since the last query.
Samples
/lx-009/systemModule
1 : 1 FreeMemory 1.472.641 Count A snapshot of free memory (in KB).
1 : 1 CPUUsageSinceServerStarted 1 Average The average CPU utilization since the server was started.
1 : 1 CPUUsageSinceLastMeasurement 17 Count The average CPU utilization since the last query.
/lx-010/systemModule
1 : 1 FreeMemory 1.519.916 Count A snapshot of free memory (in KB).
1 : 1 CPUUsageSinceServerStarted 1 Average The average CPU utilization since the server was started.
1 : 1 CPUUsageSinceLastMeasurement 9 Count The average CPU utilization since the last query.
Real-time measurements
Delta computations
Chart legend
Available statistics
Historical measurement

The real-time measurements area contains several collapsible sections. These sections are nested. On the top level you can collapse or expand the `aggregated samples` and `samples` section. Within these sections another collapsible area is available for each metric which contains the recorded details. The collapsible sections can be expanded or collapsed by clicking on the icon or double clicking on the title.

PMIExplorer automatically aggregates all similar metrics in a workspace. In the image above the `systemModule` PMI metric was monitored for two nodeagents. These recordings are show in the samples tables as `/lx-009/systemModule` and `/lx-010/systemModule`. The sum of these recordings is shown in the aggregated samples table as `/systemModule`.

A filter is available within the real-time measurements panel. This filter contains the paths of the metrics and can be used to hide tables which are less interesting.

Each real-time sample table contains the following columns:

Checkbox: Toggles plotting of the statistic on the chart

Legend icon: Shows the icon used in the chart when the statistic is plotted

Scale modifier: Allows to adjust the scale for the statistic on the chart

Statistic name: The name of the statistic

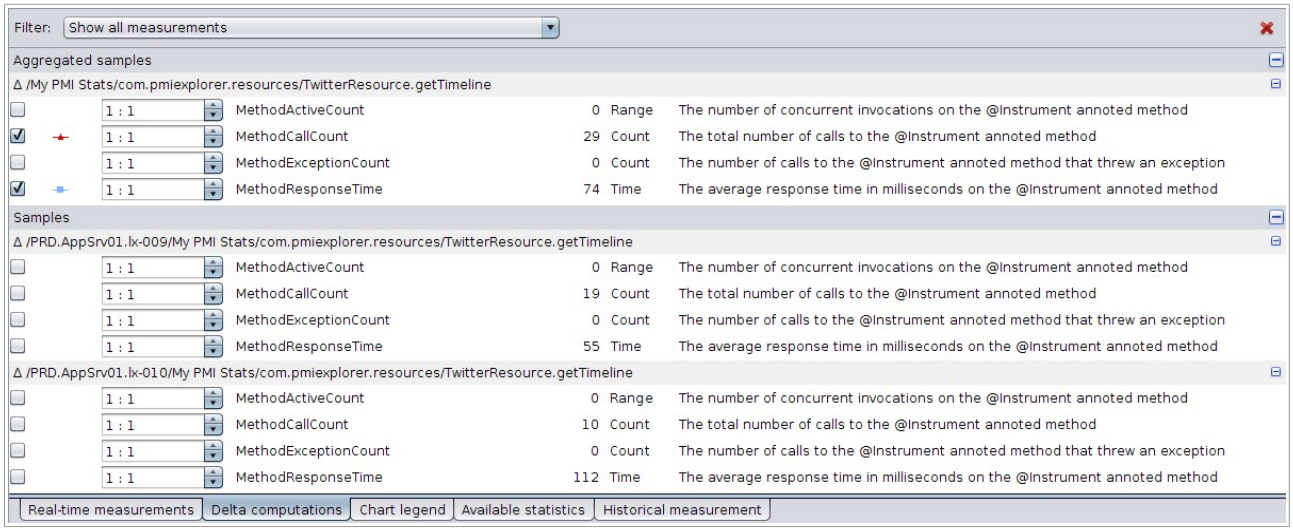
Statistic value: The actual recorded value

Statistic type: The type of statistic (count, range, time, average)

Statistic description: Description of the statistic.

Delta computations

The delta computations tab contains the details for the computed data. PMIExplorer continuously calculates the difference between two consecutive samples. By performing these calculations PMIExplorer is able to determine the real response times or calls per second on a service. The image below shows the delta computations tab:


Filter: Show all measurements
Aggregated samples
Δ /My PMI Stats/com.pmiexplorer.resources/TwitterResource.getTimeline
1 : 1 MethodActiveCount 0 Range The number of concurrent invocations on the @Instrument annoted method
1 : 1 MethodCallCount 29 Count The total number of calls to the @Instrument annoted method
1 : 1 MethodExceptionCount 0 Count The number of calls to the @Instrument annoted method that threw an exception
1 : 1 MethodResponseTime 74 Time The average response time in milliseconds on the @Instrument annoted method
Samples
Δ /PRD.AppSrv01.lx-009/My PMI Stats/com.pmiexplorer.resources/TwitterResource.getTimeline
1 : 1 MethodActiveCount 0 Range The number of concurrent invocations on the @Instrument annoted method
1 : 1 MethodCallCount 19 Count The total number of calls to the @Instrument annoted method
1 : 1 MethodExceptionCount 0 Count The number of calls to the @Instrument annoted method that threw an exception
1 : 1 MethodResponseTime 55 Time The average response time in milliseconds on the @Instrument annoted method
Δ /PRD.AppSrv01.lx-010/My PMI Stats/com.pmiexplorer.resources/TwitterResource.getTimeline
1 : 1 MethodActiveCount 0 Range The number of concurrent invocations on the @Instrument annoted method
1 : 1 MethodCallCount 10 Count The total number of calls to the @Instrument annoted method
1 : 1 MethodExceptionCount 0 Count The number of calls to the @Instrument annoted method that threw an exception
1 : 1 MethodResponseTime 112 Time The average response time in milliseconds on the @Instrument annoted method
Real-time measurements
Delta computations
Chart legend
Available statistics
Historical measurement

The delta computations area contains the same collapsible sections and filter options as the real-time measurements tab. PMIExplorer also automatically aggregates all computed statistics for metrics with the same path in a workspace. The collapsible sections can be expanded or collapsed by clicking on the icon or double clicking on the title.

In the image above the `TwitterResource.getTimeline` PMI metric was monitored for two application servers. The computed results are shown in the samples tables. The aggregation of these recordings is shown in the aggregated samples table as `Δ /My PMI Stats/com.pmiexplorer.resources/TwitterResource.getTimeline`.

The computed value for MethodCallCount of 29 means that compared to the previous sample, which was taken 10 seconds before, 29 calls were made to a specific function, which is about 3 calls per seconds.

The computed MethodResponseTime of 74 means that the average over 10 seconds (the collector interval) was 74 msec. Looking at the individual server values the computed delta shows that one of the servers was slightly slower compared to the other.

Each delta computation sample table contains the following columns:

Checkbox: Toggles plotting of the delta statistic on the chart

Legend icon: Shows the icon used in the chart when the statistic is plotted

Scale modifier: Allows to adjust the scale for the delta statistic on the chart

Statistic name: The name of the statistic

Statistic value: The computed value

Statistic type: The type of statistic (count, range, time, average)

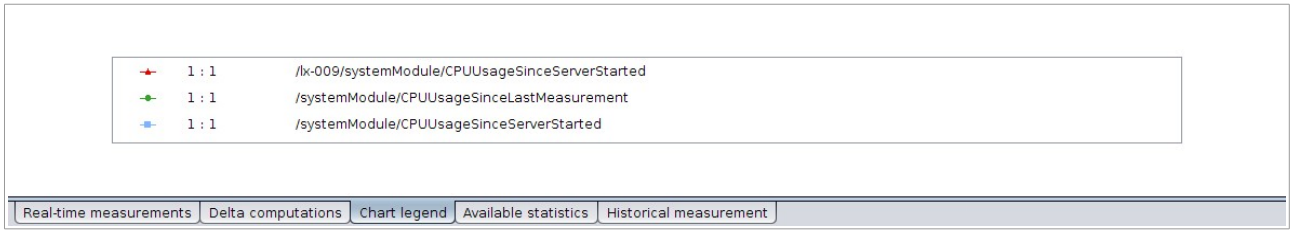
Statistic description: A description of the statistic

NOTE:

PMIExplorer always computes the delta values. The computed values are not stored in the database, but the values can be used for plotting a chart, defining an alert or exporting statistics. When you plot a chart with a coarsened granularity then the number of data point is taken into account while computing the value.

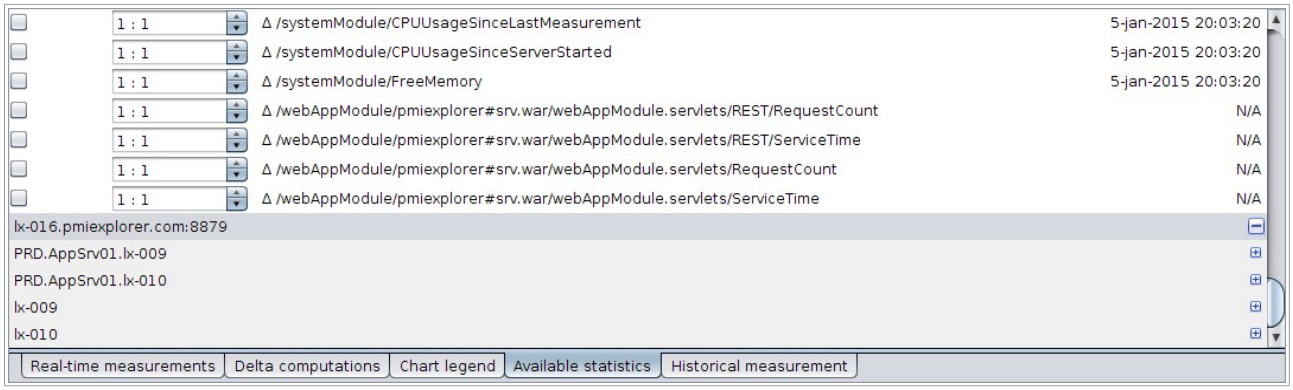
Chart legend

The chart legend lists the statistics in the chart with their corresponding shape and color. The legend also contains scale for each plotted statistic.



Available statistics

The available statistics panel shows the statistics that have been recorded at least once. This panel provides another means to view back data when collector is currently not recording samples from WebSphere.



This view is populated when an endpoint has been added to a workspace. It is not necessary to have an active connection for this endpoint. You can add endpoints from the `Add endpoint` dialog by clicking on the button `Add` instead of `Connect`.

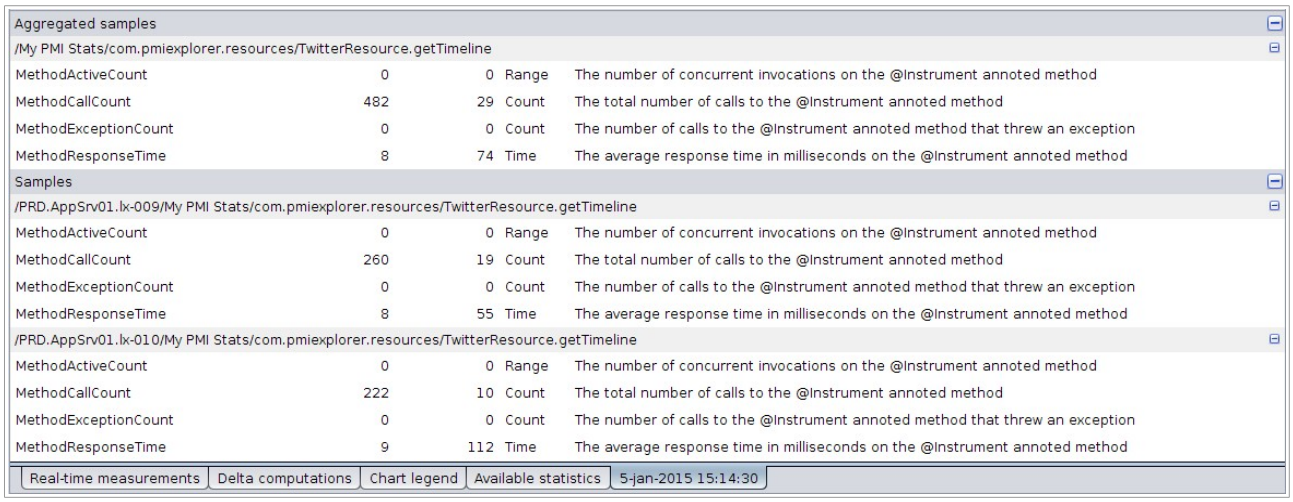
The available statistics view has collapsible sections. The top section contains the aggregated statistics. The bottom sections have a drill down from the endpoint to the server which contains the statistics. By default the bottom sections are collapsed. These sections can be expanded by double clicking on the title or clicking on the icon.

The available statistics tables contain the following columns:

Checkbox:	Toggles plotting of the (delta) statistic on the chart
Legend icon:	Shows the icon used in the chart when the statistic is plotted
Scale modifier:	Allows to adjust the scale for the (delta) statistic on the chart
Statistic path:	The full path of the statistic
Timestamp:	The last recorded or computed time of the statistic

Historical measurement

The historical measurement view contains the recorded details for one moment in time. This view is filled when clicking on the shape in the line on the chart. PMIExplorer retrieves all recorded data for that timestamp from cache or the database. Also the data for the previous timestamp is retrieved to calculate the delta values.


Aggregated samples
/My PMI Stats/com.pmiexplorer.resources/TwitterResource.getTimeline
MethodActiveCount 0 0 Range The number of concurrent invocations on the @Instrument annoted method
MethodCallCount 482 29 Count The total number of calls to the @Instrument annoted method
MethodExceptionCount 0 0 Count The number of calls to the @Instrument annoted method that threw an exception
MethodResponseTime 8 74 Time The average response time in milliseconds on the @Instrument annoted method
Samples
/PRD.AppSrv01.lx-009/My PMI Stats/com.pmiexplorer.resources/TwitterResource.getTimeline
MethodActiveCount 0 0 Range The number of concurrent invocations on the @Instrument annoted method
MethodCallCount 260 19 Count The total number of calls to the @Instrument annoted method
MethodExceptionCount 0 0 Count The number of calls to the @Instrument annoted method that threw an exception
MethodResponseTime 8 55 Time The average response time in milliseconds on the @Instrument annoted method
/PRD.AppSrv01.lx-010/My PMI Stats/com.pmiexplorer.resources/TwitterResource.getTimeline
MethodActiveCount 0 0 Range The number of concurrent invocations on the @Instrument annoted method
MethodCallCount 222 10 Count The total number of calls to the @Instrument annoted method
MethodExceptionCount 0 0 Count The number of calls to the @Instrument annoted method that threw an exception
MethodResponseTime 9 112 Time The average response time in milliseconds on the @Instrument annoted method
Real-time measurements
Delta computations
Chart legend
Available statistics
5-jan-2015 15:14:30

The historical measurement view also contains collapsible sections and a filter to look up a specific metric. The top section contains the aggregated samples and the bottom collapsible sections contains the values recorded per server.

The tables contains the following columns:

Statistic name: The name of the statistic

Real-time value: The value recorded from WebSphere

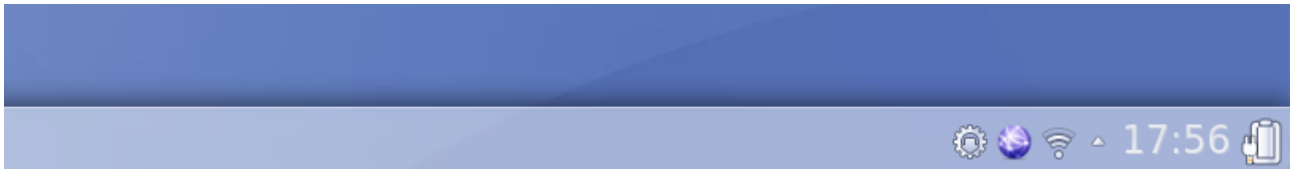
Delta computation: The value computed by PMIExplorer

Statistic type: The type of statistic (count, range, time, average)

Description: The description of the statistic

System tray

PMIExplorer adds an icon to your system tray when the operating system supports it. Right click the tray icon to access the pop-up menu which gives access to alert filtering options and has some other controls. We recommend to configure your operating system to always show the icon, because the icon is updated with a warning symbol when an alert is fired.



Alerts

When PMIExplorer generates an alert this alert will be displayed on the desktop in a pop-up in the lower right corner of the desktop. This pop-up contains some details about the

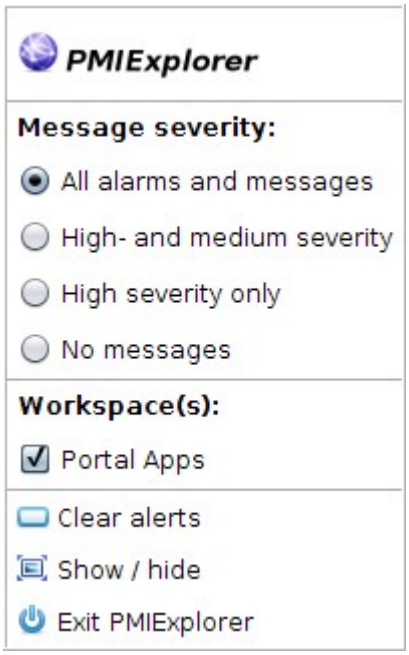
alarm including the value which triggered the alarm. You can also display the pop-up by clicking on the PMIExplorer task tray icon or hovering on it for one second.

The pop-up shows the most recent alerts where the severity is indicated by an icon. Also the name, workspace and last recorded value value are shown in the message. The alarm can be dismissed by clicking on the red cross after the alert. The following symbols indicate the severity of the alarm:

- High
- Medium
- Low

Filters and controls

A right mouse click on the tray icon opens the pop-up menu where you can set severity filters for the alarms. Select the appropriate option to suppress alerts with certain severity or enable / disable them for a workspace. The other actions available in the menu can clear the list of alerts, minimize PMIExplorer to the task tray or exit the program.



Connecting to WebSphere

Connection context

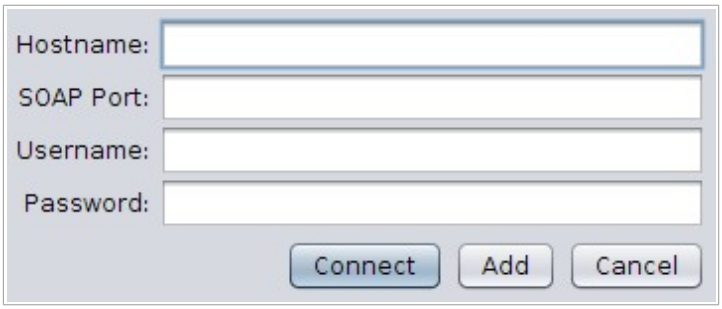
In order to retrieve PMI data from WebSphere you need to connect to a WebSphere endpoint. This endpoint can be a deployment manager for distributed environments or a standalone server. It is also possible to connect to an individual server in a distributed environment, but it's recommended to connect to the deployment manager since the connection to the DMGR is kept alive while individual servers in a cell might be restarted.

PMIExplorer uses the WebSphere AdminClient connector and mbean technology to retrieve the PMI data. A secure connection using SSL certificates is required. By default PMIExplorer automatically accepts the certificates presented on the connection, but it is possible to configure the keystore manually. To accept only the (trusted) certificates in the keystore you need to set `pmi.ssl.automatic` to false.

When running PMIExplorer with LDAP security the directory server certificate is also accepted automatically or otherwise it has to be imported into the keystore too. See `Setting up the keystore` in the chapter `Security` for more details on configuring the keystore.

Creating the connection

On the button bar there is a button `Add endpoint` which opens the WebSphere connection dialog.



In this dialog the following information is required to create the connection:

Hostname:	The full name, hostname or IP address of the WebSphere endpoint
SOAP Port:	The SOAP port number; usually 8879 (dmgr) or 8880 (standalone)
Username:	A user account which is authorized to view the PMI data
Password:	The password for the user

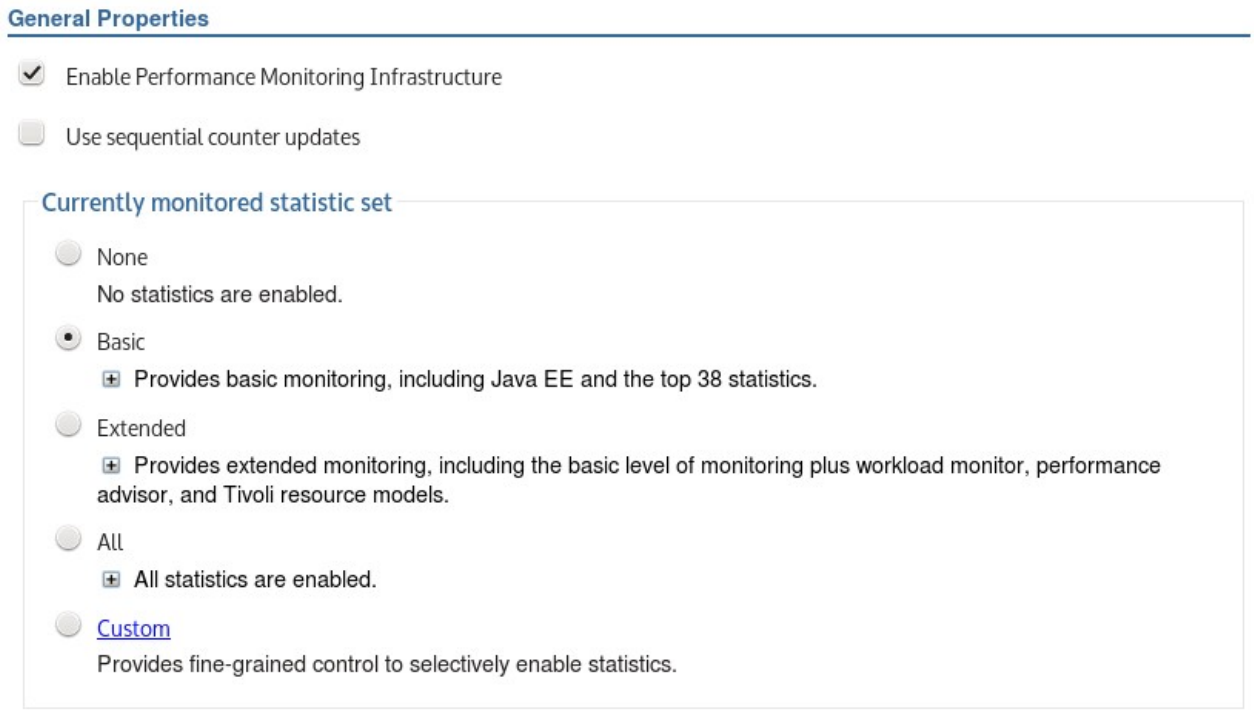
In this dialog there are two buttons available. The button `Connect` establishes an AdminClient connection. Only under this condition the PMIExplorer can retrieve data from WebSphere. The button `Add` adds the endpoint to the workspace and does not establish a connection, which is useful for viewing historical data.

PMIExplorer leverages the security roles from WebSphere. This means that in order to record the PMI data the login must at least have the **monitor** privilege in the admin console. When running PMIExplorer as remote agent it is recommended to create a special account in WebSphere, grant the monitor role and not change the password frequently. PMIExplorer will loose it's connection after the password has been changed.

When you connect to a WebSphere endpoint on a remote agent you are transmitting the username and password from the client to the agent. PMIExplorer does not use SSL for client to agent communication, but all sensitive information, including usernames and passwords, is encrypted with an AES cipher to keep the communication secure and the overhead low.

Configuring PMI in WebSphere

Which metrics are available to PMIExplorer is determined in the WebSphere configuration. This configuration is available in the admin console under `Monitoring and Tuning` → Performance `Monitoring Infrastructure (PMI)`. Out of the box WebSphere provides the following presets:


General Properties
Enable Performance Monitoring Infrastructure
Use sequential counter updates
Currently monitored statistic set
None
No statistics are enabled.
Basic
Provides basic monitoring, including Java EE and the top 38 statistics.
Extended
Provides extended monitoring, including the basic level of monitoring plus workload monitor, performance advisor, and Tivoli resource models.
All
All statistics are enabled.
Custom
Provides fine-grained control to selectively enable statistics.

By customizing these settings you can enable more metrics and / or more statistics per metrics. It is possible to define a fine grained statistic configuration by customizing the WebSphere configuration.

Adjustments can be made in the server runtime, which makes the metrics available for PMIExplorer immediately. It is possible in the WebSphere admin console to persist the runtime changes to the master configuration.

NOTE:

Since WebSphere 8 sometimes the PMI statistics are not enabled. Verify your WebSphere configuration if you are able to connect to WebSphere but do not see available metrics.

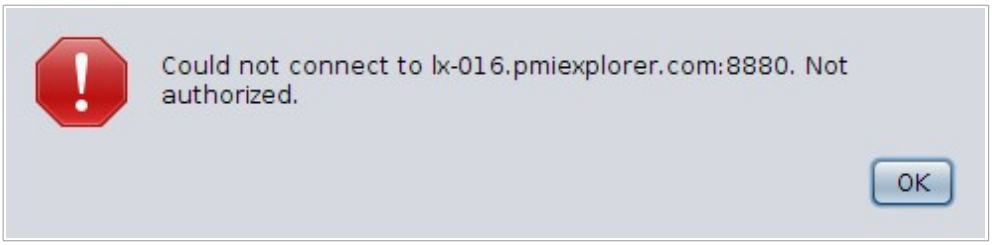
NOTE:

When additional metrics are added to WebSphere, like for WebSphere Process Server or our Instrumentation API then these metrics do not show up in on the configuration page when customizing the configuration. The only way to modify the configuration for these metrics and reload this configuration after a server restart is customizing the runtime and persisting the changes from there.

Common connection problems

Authorization failure

This condition might when the username of password was mistyped, the account was disabled, password was expired or a back-end authentication source is not available. Also when the user was not granted the `monitor` role in the admin console an authorization failure is shown.

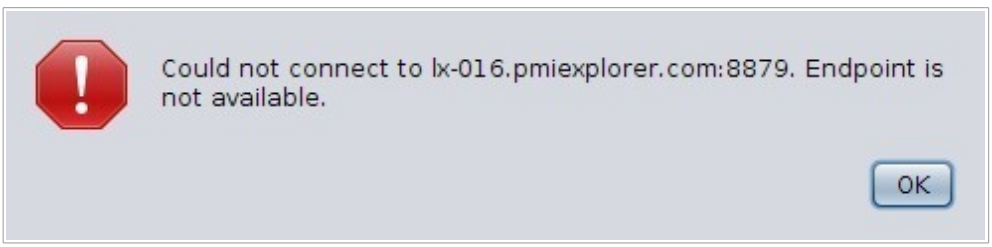


To resolve this error you could perform the following diagnostic actions:

1. Verify the credentials and try again
2. Verify if the user can log on to the admin console
3. Verify if the users password has expired or the account is locked
4. Verify if the authentication source is still available.

WebSphere not available

This condition occurs when the WebSphere deployment manager or standalone server is not available or a wrong hostname or port was entered while connection. Also network and firewall issues might lead to this same issue.



To resolve this error you could perform the following diagnostic actions:

1. Verify the hostname and port
2. Verify if the deployment manager or standalone server is running

3. Attempt to connect to the SOAP connector port via telnet
4. Check if other ports on the same (virtual) machine are available
5. Check for SSL related errors in the PMIExplorer and WebSphere logs

SSL certificate related issues

This condition might occur if certificates are not accepted automatically. If the `pmi.ssl.automatic` property has been disabled than PMIExplorer does not establish a connection when a certificate is not present in the trusted (signers) section of the keystore. This will lead to a connection error in the logs. This also applies to LDAP connection.

To resolve this error you could perform the following diagnostic actions:

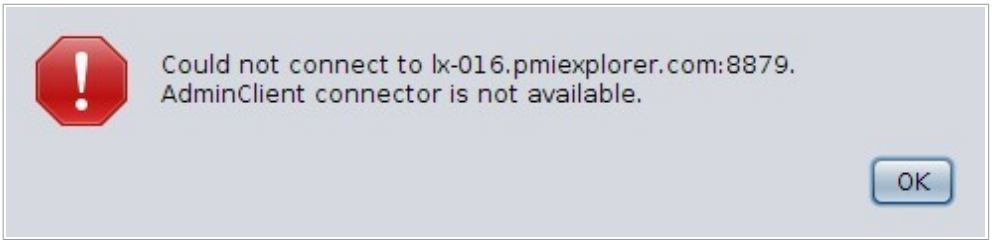
1. Verify if the `pmi.ssl.automatic` property has the right value
2. Verify the certificate on the SOAP port by accessing it with a web browser
3. Verify if the certificate is present in the signer section of the keystore
4. Verify expiration dates and serial numbers match
5. Verify access to the PMIExplorer keystore and examine the log carefully

NOTE:

The deployment manager can automatically regenerate the certificates depending on your configuration. When this renewal happens than PMIExplorer will loose the connection to the deployment manger, because this certificate regeneration is not automatically propagated to the PMIExplorer keystore. In this case you must add the renewed deployment manager certificate to the keystore again.

Connector jars not available

This condition occurs when the `com.ibm.ws.admin.client-x.x.x.jar` and `com.ibm.ws.security.crypto.jar` are not available in the ext directory. Without these libraries PMIExplorer can not access the WebSphere server(s).



To resolve this error you could perform the following diagnostic actions:

1. Verify if the libraries are available in the ext directory
2. Verify the file permissions on the libraries
3. Check the log file for error messages related to library loading

PMI data collector

PMI data in PMIExplorer

Path expressions

Within PMIExplorer we use several terms which indicate how data is structured. To uniquely identify a statistic a path can be constructed for each statistic which is based on a tree structure.

Endpoint: An endpoint represents the connection to WebSphere. The endpoint gives access to one cell in WebSphere. An endpoint contains one or more servers.

Server: A server translates to a JVM in WebSphere. PMI configuration in WebSphere applies to one server. A server contains zero or more metrics. Metrics can be nested in groups with a similar function, like Java beans or servlets.

Metric: A metric is a performance indicator which relates to a component or resource in WebSphere like a connection pool or servlet container. WebSphere also contains groups of metrics in the tree. A metric has performance indicators (statistics). Metrics can have a nested group structure.

Statistic: A statistic is a performance indicator for a metrics for which the samples are recorded. You may require more than one statistic to make statements about the performance of a component (metric).

Delta: The capital Greek delta sign (Δ) is used to indicate if the statistic represents a real-time (or recorded) statistic or a computed delta statistic.

To uniquely identify a statistic within PMIExplorer a path structure is used. This path consists of the following components:

delta / server / metric group(s) / metric / statistic

The delta sign indicates if the path identifies a computed statistic. When the sign is not present then the path represents a real-time statistic. The `server` in the path is optional. When this server is not present in the path then the path represents an aggregated statistic (combined statistic for multiple servers). The example below shows the path for a computed statistic on `server1` which refers to the `MethodResponseTime` for the `MyApplicationMDB`

```
Δ /server1/beanModule/ejb.MessageDriven/MyApplicationMDB/MethodResponseTime
```

Statistic types

JSR-77 specifies several types of statistics. These statistic types are also available in WebSphere and can be read by PMIExplorer. The following statistic types are used in WebSphere.

CountStatistic

The count statistic is used to record the amount of calls on a metric. This type of statistic holds the number of calls since the WebSphere server was started and is normally increased by 1 when the component is invoked.

RangeStatistic

The range statistic is commonly used to record the current value of a metric, like usage of a connection pool or the CPU utilization. The JSR-77 specification also defines a bounded range statistic, which is treated as a regular range statistic by PMIExplorer. High- and low watermarks are therefore not available in PMIExplorer.

TimeStatistic

The time statistic records the time it took to execute a component (i.e. method or connection usage). WebSphere shows the average since server startup when using the Tivoli Performance Viewer.

AverageStatistic

The average statistic is used to create the mean value for a (bounded) range statistic. This type of statistic is used rarely in the WebSphere stats.

Real-time recordings

PMIExplorer takes real-time recordings from WebSphere by invoking the PerfMBean in WebSphere which gives access to the performance data. By default the collector which runs as part of PMIExplorer reads the values from the PerfMBean every 10 seconds, depending on the `pmi.collector.interval`.

These values are stored in the database and are used to view back historical data. The data is written to the database asynchronously, using an intermediate buffer, which ensures the collector can continue reading PMI data from WebSphere every 10 seconds.

The real time recordings are relayed to all clients connected to a remote agent. The clients update their details panels and calculate the delta to the previous sample when they receive an event from the remote agent. PMIExplorer has a delay of one interval (10 seconds by default) for sending alerts based on the recordings.

Delta computations

For every real-time statistic a delta statistic is computed to the previous recording. This derived statistic adds great value to your WebSphere PMI data and gives a new dimension to the PMI data. These deltas are most useful for the count statistic and the time statistic.

Count statistic

By default WebSphere gives you the value since server startup. When a server has been running for weeks or months you will only see a steady increase when plotting the statistic on the chart. The derived count statistic (delta) for the same metric value represents the transactions per second. Depending on the granularity used in the chart you can view the delta statistic as transactions per 10 seconds (default collector interval), transactions per minute, or transactions per hour.

The computed delta statistic is most useful in alerting when monitoring application throughput. You could raise an alarm when the throughput (TPS) drops below a certain expected value based on the delta statistic.

Time statistic

By default the Tivoli Performance Viewer in WebSphere shows the time statistics as the average since server startup. When an application server is running for days, weeks or months a deviation in the response time hardly has any effect on this average value.

Determining the actual value responsible for a deviation based on the server average since startup is not possible. The derived delta time statistic computes the response time over a shorter interval (10 seconds when using the default collector interval). In this case the derived value represents the real-time performance for an application.

The computed delta time statistic are therefore also of priceless value in performance analysis or alerting. It is possible to use these computed delta values to sound an alarm when response times for a certain service are increasing above a threshold.

Aggregated statistics

The aggregated statistic is an additional computation on the recorded data. Aggregation is applied to both the real-time measurements and computed delta statistics. When data from multiple nodes is aggregated that creates a new statistic. Within a workspace metrics with the same path are automatically aggregated.

The aggregation of data is most useful for applications that run on a WebSphere cluster or are installed on multiple standalone servers. The newly created aggregated statistic reflects the performance for the nodes together. The details from each node are still available and can be used to check if one node deviates from the others, which might indicate a problem specific to that node.

NOTE:

In some cases the aggregated statistic cannot be computed for the delta statistics of the nodes in a cluster. To accurately calculate the aggregated delta statistic there must be an equal amount of data points to perform the computation. When stopping and restarting an application server the number of data points to aggregate are temporarily not equal and at that moment it is not possible to reliable calculate the aggregated delta statistic.

Sizing the database

Storing PMI data in a database is going to require physical storage. How much storage you need depends on several factors like redundancy, which may be available on database level (like Oracle RAC or DB2 HADR) or disk pool (RAID configuration) and backup.

Sizing the sample table

The sample table will become the largest table in PMIExplorer database and contains most records. The amount of records is determined by several variables:

- Number of recorded metrics
- Number of statistics per metric
- The collector interval (defaults to 10 seconds)
- The collector retention (defaults to 30 days)
- The size of the sample record, which is approximately 28 bytes

The following formula can be used to make an estimation of the database size required for your monitoring environment.

```
(retention * 86400 / interval) * monitored metrics * stats per metric * record size
```

Average monitoring environment

Considering the following monitoring environment:

Retention : `30 days`

Interval : `10 seconds`

Monitored metrics : `30`

Stats per metric : `4`

Using the sizing formula this would give an approximate size for the sample table:

```
(30 * 86400 / 10) * 30 * 4 * 28 bytes  =  870.912.000 bytes (831 MB)
```

Large monitoring environment

Considering the following monitoring statistics (for a larger monitoring setup):

Retention : `100 days`

Interval : `10 seconds`

Monitored metrics : `600`

Stats per metric : `5`

Using the sizing formula this would give an approximate size for the sample table:

```
(100 * 86400 / 10) * 600 * 5 * 28 bytes  = 72.576.000.000 bytes (68 GB)
```

NOTE:

As shown in the calculations the SAMPLE table will contain a lot of records. Depending on your database system and organizations standards you might want to implement several partitions to store the data. When choosing a partitioning key we recommend including SAMPLE_TIME because most query's use that column. For big databases you might even consider sub partitioning using the STATISTIC column.

Other tables and indexes

Other tables in the PMIExplorer database will grow to several Megabytes. This size depends mostly on how many endpoints are connected and the amount of metrics on these endpoints.

Indexes are harder to size, because this depends on the database vendor specific algorithms. The most important index in the `SAMPLE_TIME_IDX` which indexes two columns to give fast access to the data. This index size is approximately 80% of the `SAMPLE` table size.

Since a historical query attempts to retrieve data between two points for one statistic the index is often scanned. The bigger the index becomes the slower the performance, because the database needs to scan through more index entries.

NOTE:

Rebuilding the index periodically (weekly or monthly) prevents wasted storage and increases the PMIExplorer performance when retrieving historical data. Refer the you database documentation for advanced options on index creation and reorganization.

Monitoring metrics

Starting and stopping monitoring

You can start or stop monitoring a metric from the tree panel which shows the endpoints, servers and metrics. You are allowed to start and stop monitoring when you have the `editor` role in PMIExplorer. The `viewer` role is not allowed to make changes to the monitoring configuration, but does see the current state of the monitoring:

- The metric is available for monitoring
- The metric is currently monitored
- The metric monitored as part of a monitoring profile

When you click on a metric in the tree then this metric is registered to the collector component which starts recording data for that metric. Clicking the metric again instructs the collector component to stop recording performance data for this metric. Metrics that are part of a monitoring profile are always recorded. To stop monitoring for these metrics you need to remove the monitoring profile. Only then the metrics responds to the clicks again.

Stop all monitoring

On the button bar there is one button which stops all monitoring. When you click on the `Stop monitoring metrics` button then you need to confirm the action to halt all monitoring to avoid stopping the monitoring by accident. When you stop all monitoring then this action also removes all active monitoring profiles from the workspace, which would otherwise prevent the stop monitoring action.

NOTE:

Use this function carefully, because this will halt recording data. You could save the current monitoring state to a profile if you think you need to restart monitoring later.

NOTE:

Stopping all monitoring might trigger an alarm if you configured alerts for metric unavailability. Throughput alerts are not triggered, because there is no data to calculate the throughput.

Monitoring profiles

A monitoring profile contains a group of metrics, which might be from different servers, which can be stored to a file and shared with other installations of PMIExplorer. This monitoring profile is used to register multiple metrics to the collector at once and provides a safeguard against accidentally disabling monitoring for a metric.

During detection of new servers and metrics for an endpoints the monitoring profile is used to automatically start monitoring for metrics that are part of the monitor set. When your restart a WebSphere server then the metrics are temporarily unavailable, but once they become available again then monitoring is resumed automatically.

The PMIExplorer remote agent can remember the monitoring state when it is restarted. During startup the agent reloads the endpoints and starts the monitors based on the monitoring profile(s). The standalone client does not automatically reload the endpoints and monitoring profiles.

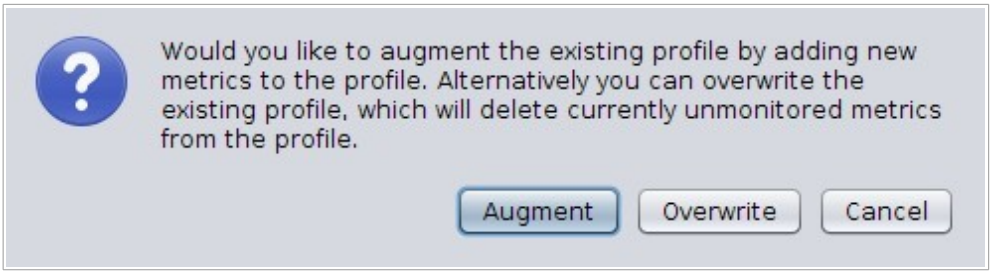
To create, load and unload monitoring profiles you must have at least `editor` privileges in PMIExplorer. Users in the `viewer` role are not allowed to make changes to the monitoring configuration.

Creating monitoring profiles

Use the button `Save monitoring profile` to create a monitoring profile. When you click the save button a file dialog opens which asks you for a location and file name to store the profile. By default PMIExplorer stores the profiles in the `profile` directory where PMIExplorer is installed.

When you save a monitoring profile all metrics currently selected on the tree are saved in a `.pmp` (PMIExplorer monitoring profile) file. When you would like to save metrics from multiple servers in a profile you need to enable monitoring for all these metrics on that server. PMIExplorer has chosen this approach because the approach to automatically start monitoring metrics without taking the server into consideration would result in starting monitors which might not be intended. Monitoring the heapsizes in a clustered environment for example would monitor the heapsize of the application servers and the nodeagent where one might not be interested in the nodeagents behavior (which automatically becomes part of the aggregated statistics).

When attempting to overwrite a previously created monitoring profile a dialog is opened which requires a choice between overwriting and augmenting.


Would you like to augment the existing profile by adding new metrics to the profile. Alternatively you can overwrite the existing profile, which will delete currently unmonitored metrics from the profile.
Augment
Overwrite
Cancel

When choosing the `Augment` option PMIExplorer first reads the current profile and than adds the metrics that are not yet part of the monitoring profile. When an endpoint is not

available in the workspace its metrics are not monitored. Choosing the `Overwrite` option would result in this case result in that metrics not being part of the monitoring profile anymore.

NOTE:

Metrics might not be available in PMIExplorer shortly after a WebSphere AppServer restart, because the PMI data is initialized on first usage. Special care is required when creating profiles that include servlets, method level data (from instrumentation API) or BPEL flow data in the case of WebSphere Process Server.

Loading monitoring profiles

Use the button `Load monitoring profile` to load a previously save profile. When loading the monitoring profile this will start monitoring for all metrics that are available in the profile. On the profiles section on status bar will be updated with the newly enabled profile.

When the profile is loaded the metrics in the tree will be updated with white icons which indicate that the metric is part of a profile. It is possible that metrics are part of multiple profiles. It is not possible to stop monitoring for a metric which is part of a profile. You must first disable the profile before you can stop monitoring a metric.

Disabling monitoring profiles

You can disable a monitoring profile by clicking on the button on the status bar.

The image above shows the part of the status bar which contains two profiles where one profile contains the system metrics and the other contains the application metrics. When the `'System'` profile button is clicked then PMIExplorer disables the monitoring profile for the related metrics, unless the metric is also part of another monitoring profile.

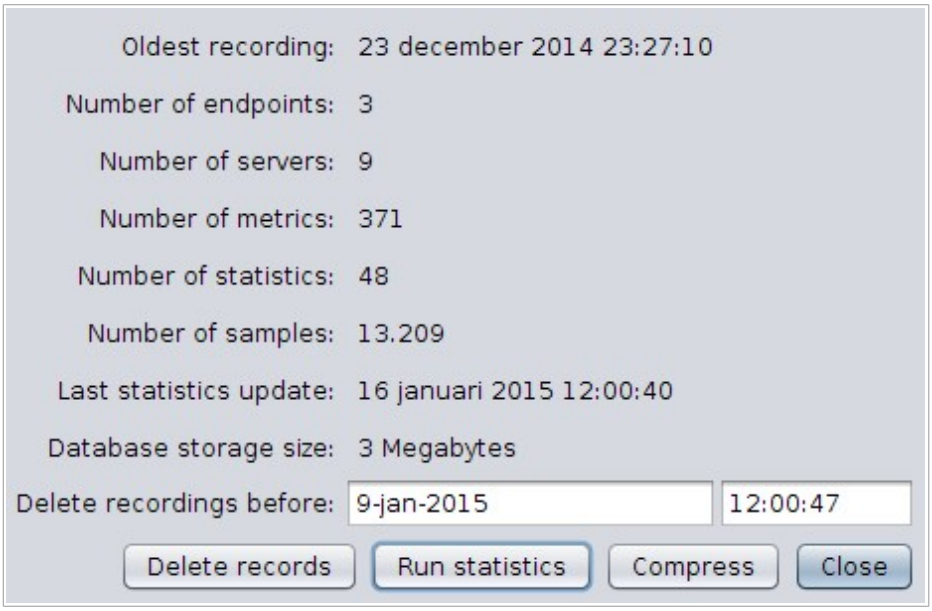
These metrics will still be monitored. Disabling a monitoring profile does not stop monitoring of a metric. The metrics that were part of the monitoring profile will turn green in the tree.

To stop monitoring you need to click on the metric in the tree or use the `Stop all monitoring` button on the button bar.

Data maintenance

Manual database maintenance

The button `Database maintenance` gives access to the database statistics, purging controls and compression option. You can also invoke the run statistics statistics task from here. The options to purge and compress data or collect statistics on the database are only available when you have the administrator role in PMIExplorer. When you click on the button the following dialog opens:



The dialog provides insights in the size of the database and amount of records in the database. The oldest recording is listed and there is a button available to `Delete records` which are older than the specified date.

The button `Run statistics` invokes a task on the database to gather the statistics. Up-to-date statistics help in achieving the optimal performance when viewing historical data. Normally the statistics task is scheduled on the database, but when the statistics are stale you can invoke a manual run.

The button `Compress` is only available when you run on the internal database which uses the Apache Derby / JavaDB database engine. Compressing data removes unused space which remains in the database files after deleting records and optimizes the storage usage.

Automatic database maintenance

PMIExplorer also has an automatic database maintenance task. This task runs daily at midnight and removes records which are older than the configured with the `pmi.collector.retention` parameter. This task also invokes the compression job when you run PMIExplorer on the internal database. The automatic record pruning task can be disabled by setting the collector retention to zero or a negative value

If you want to invoke special logic during the nightly database maintenance cycle you can write your own stored procedure and have PMIExplorer invoke that procedure. For PMIExplorer to invoke you procedure you must store it in the same schema as the PMIExplorer data and the procedure must take one argument; an epoch / UNIX timestamp, which indicates that the procedure should delete all records before this time.

If you do not run your statistics nightly on your database you can also invoke the job to gather the statistics during the PMIExplorer database maintenance cycle. In order to invoke the task on your database you must set `pmi.jdbc.runNightlyStatistics` to `true`.

Other database maintenance tasks

Depending on your database system maintenance plans you might need additional actions to hold on to a good PMIExplorer performance. As calculated in the `Sizing the database`

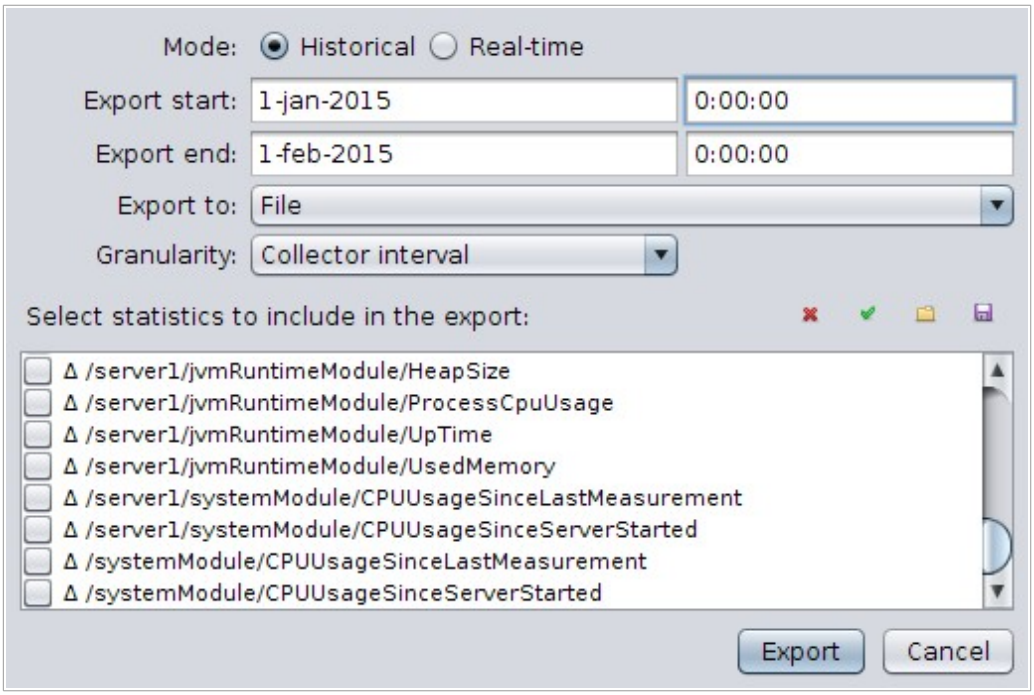
paragraph the SAMPLE table might contain lots or records. During the maintenance PMIExplorer normally deletes these records using JDBC.

Some effects have been observed on the indexes which kept on growing and caused deteriorating performance. PMIExplorer therefore recommends to periodically reorganize or rebuild your indexes or implement other measures like partitioning with local indexes, which can be dropped when the stored procedure is invoked.

PMIExplorer deletes data in small blocks (number of samples collected in 5 minutes) to avoid impact on database transaction management system caused by PMIExplorer also inserting new records every 10 seconds (default collector interval). More efficient record pruning might be implemented by writing a stored procedure (which takes a Unix timestamp as input) to clean old records, which must be located in the same schema as the data and is invoked nightly when the `pmi.jdbc.pruneProcedure` parameter is configured.

Exporting data

Click on the `Export statistics` button to open the export dialog. PMIExplorer allows to export recorded data, including the delta and aggregated statistics. You can export historical data to a csv file for management reports or export the data real-time to another system where it can be used for event correlation.



Historical

The historical data export enables you to export recorded- and computed statistics to a file. PMIExplorer generates a csv file which can be used in other applications like Excel to generate management reports. The dialog contains the options to select the granularity which determines the amount of data points in the export. In the export dialog you can

check the boxes for the statistics which need to be included in the export. It is also possible to save the selected statistics to a profile. Loading the profile checks the check boxes which have been stored in the profile. The export profile enables you to consistently export the same statistics each week or month.

Real-time

Real-time exports can be used in a chain of systems involved in monitoring. The real-time export enables you to perform event correlation with the computed or recorded data. Consider the situation where PMIExplorer monitors an enterprise service bus and another monitoring product monitors a portal where the user interaction takes place. If the other system cannot compute the PMI data (applies to most monitoring systems) you can use the real-time export to feed the information to that system where it can be used together with other samples.

PMIExplorer has two embedded exporters. One exporter writes the data to system console where it can be picked up by another application and the other exporter writes the data to the PMIExplorer database. Depending on the granularity new records are inserted into the `EXPORT` table frequently, where it can be read by other applications. The records in the table are kept for 5 minutes before they expire and the exporter removes the data.

An extension point in PMIExplorer allows to create custom exporters which can be used to push the data to another system if that system has an API to receive the data.

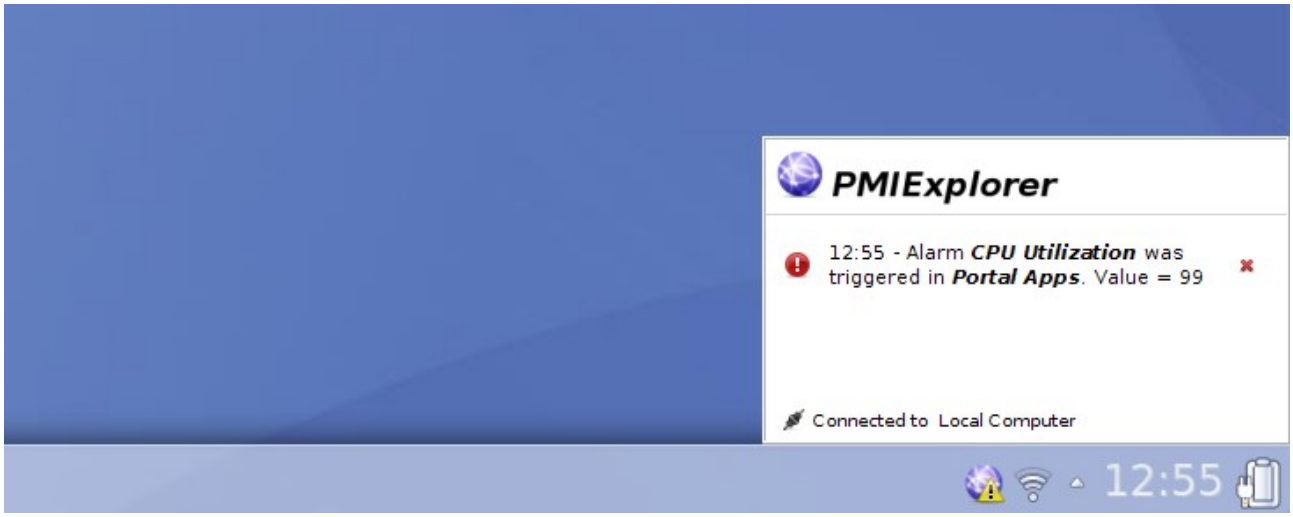
Alerting

Alerting mechanisms

Being able to broadcast alerts is a key feature of PMIExplorer. PMIExplorer has several mechanisms to alert you when a recorded or computed value exceeds a configured threshold. Alerting capabilities are provided for standalone client installations and remote agent installations.

Pop-up on the desktop

When an alert is raised from PMIExplorer then this will always result in a notification on the client. An alert will pop-up in the corner of the desktop signaling the user that an alert was raised.



In case of a remote agent installation then the alarms are generated from the agent and the alert is relayed to all connected clients. When the client receives the signal then it will show a pop-up message on the desktop and the icon in the task tray is updated with a warning symbol. When you click on the task tray icon or hover over it with your mouse that will show the recent alerts. When you click on the red cross the alarm is removed. In the case there are no connected clients then nobody will receive direct a signal.

The pop-up is always part of another alerting mechanism which ensures the signal gets the proper attention. To control the alerts on your desktop you can right click on the task tray icon. From the context menu you can configure the severity for which the pop-up is displayed on the desktop:

- All alarms and messages
- High- and medium severity
- High severity only
- No messages

The same context menu can also disable the alerting for specific workspaces and has an option to clear all alerts, which resets the task tray icon to its default state.

alarms.log

The alarms.log file is one of two alerting options which come with PMIExplorer by default. This alarm does not give direct feedback to the user, but is very useful for recording all threshold transgressions. This alert mechanism can be used when tuning the alert threshold or analyzing performance issues for which no direct interaction is required.

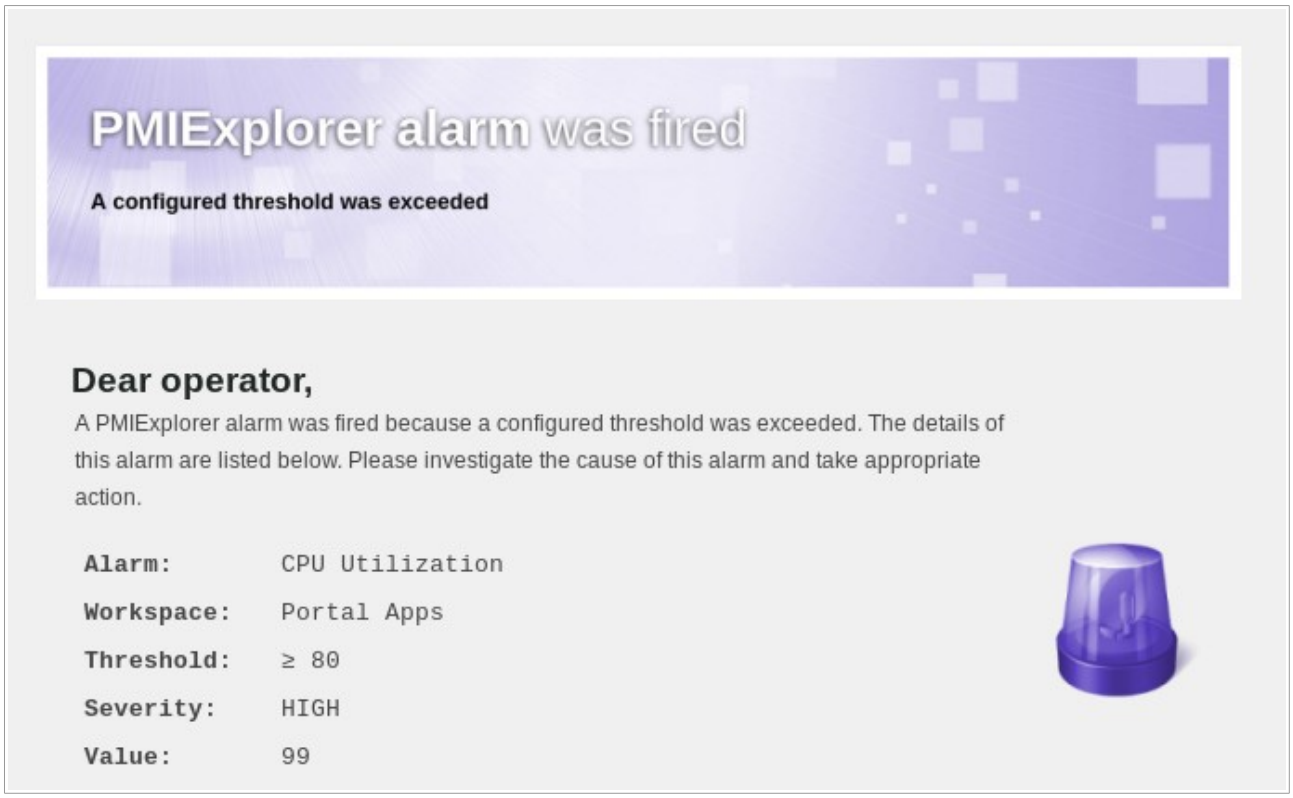
The alarms.log file is saved in the logs directory for your PMIExplorer installation and could contain records like:

```
[17-01-2015 12:00:40.002 CET] HIGH    Alarm 'CPU Utilization' was fired in workspace 'Portal Apps'.
Measurement(s) meet alarm criteria (value ≥ 80). Current value = 82.

[17-01-2015 12:55:40.002 CET] HIGH    Alarm 'CPU Utilization' was fired in workspace 'Portal Apps'.
Measurement(s) meet alarm criteria (value ≥ 80). Current value = 99.

[17-01-2015 13:24:20.001 CET] HIGH    Alarm 'CPU Utilization' was fired in workspace 'Portal Apps'.
Measurement(s) meet alarm criteria (value ≥ 80). Current value = 87.
```

E-mail

Sending the alerts over SMTP is the other option that is embedded in PMIExplorer. To use this type of alert you need to configure the `Notifications` properties in the PMIExplorer configuration file, because otherwise PMIExplorer does not know where to deliver the alarm.

When this type of alert is fired then the operators receive a notification in the configured mailbox:

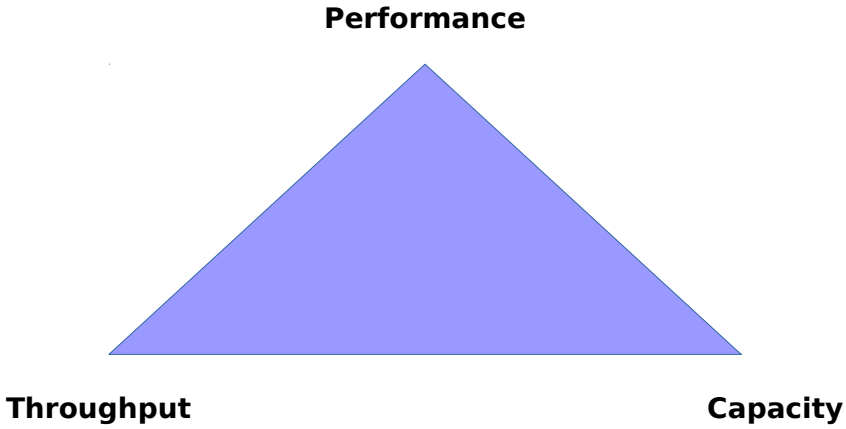

PMIExplorer alarm was fired
A configured threshold was exceeded
Dear operator,
A PMIExplorer alarm was fired because a configured threshold was exceeded. The details of this alarm are listed below. Please investigate the cause of this alarm and take appropriate action.
Alarm: CPU Utilization
Workspace: Portal Apps
Threshold: ≥ 80
Severity: HIGH
Value: 99

Customized alarm

PMIExplorer has an extension point for alerts which also allow to connect to all kinds of incident management systems and raise a ticket for the operators to investigate. PMIExplorer.com can develop such extensions for you on request.

Areas of interest

When monitoring the performance of your applications there are three common things to monitor for which are closely related:



Performance

The performance of a system is measured in the time it takes to execute a certain function. The corresponding type of statistic is a time statistic. In PMIExplorer the real-time performance can be measured through the delta computations on two samples. These computations show the performance in the timespan between the recorded samples.

The performance is often influenced by the throughput, since most applications do not scale 100% linear. When the number of transactions or invocations per second increases then the performance often decreases; the response times get higher.

The performance has an influence on the available capacity. The capacity is often limited by the amount of threads in a threadpool, connections in a connection pool or sessions on a message driven bean. When it takes more time to handle a function call then the amount of calls on that function can be limited by the maximum capacity, since there wouldn't be enough resources (threads, connections, CPU, etc.) to handle the function call

The most common performance indicators are:

- Δ MethodResponseTime (on a bean or method via the Instrumentation API)
- Δ UseTime (on a JDBC or JMS connections)
- Δ ServiceTime (on a servlet)

Throughput

The throughput of a system is measured by the amount of transactions or invocations it processes in a given time (normally TPS). The corresponding type of statistic is a count statistic. In PMIExplorer the throughput can be measured through the delta computations on two samples. The computed value show the amount of calls in a given time, which is determined by the granularity in the chart or collector interval when monitoring in real-time.

The throughput is influenced by the performance and capacity of a component. When the response times increase then the maximum throughput may be limited by the available capacity. If there are no free connections available in a connection pool for example it takes longer before a connection is available which limits the throughput.

The most common throughput indicators are:

- Δ MethodCallCount (on a bean or method via the Instrumentation API)
- Δ ReturnCount (on a connection pool)
- Δ RequestCount (on a servlet)

Capacity

The (available) capacity of a system is measured by the amount of room there is for additional load. In PMIExplorer the capacity can generally be measured by looking at the real recorded value, which is a range statistic. This statistic indicates how much of the system capacity is used, but does not always tell the system limit, which is defined in the WebSphere configuration settings or available system resources.

The available capacity relates to the throughput and performance indicators which together determine how much room there is for additional load.

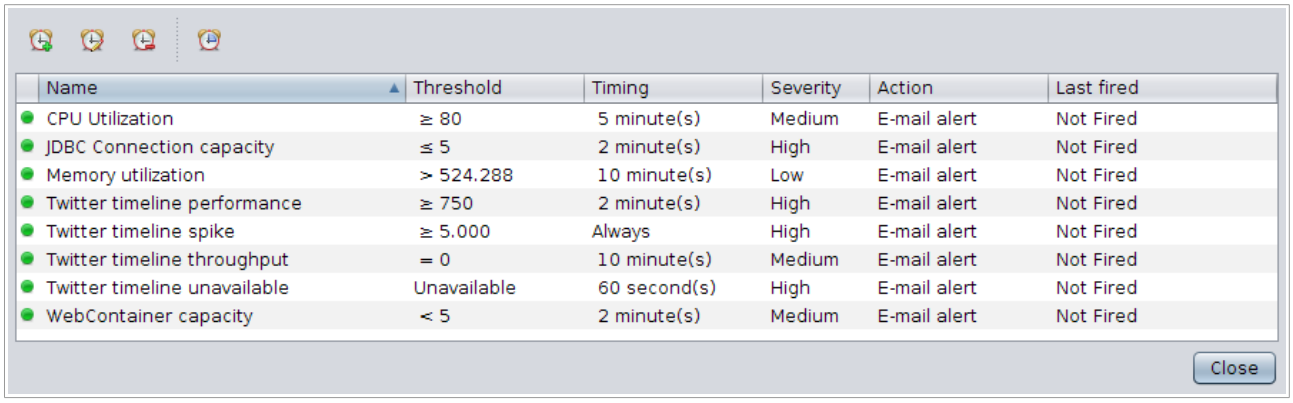
The most common capacity indicators are:

- PooledCount (for message driven beans)
- ActiveCount (on threadpool like for the WebContainer)
- ConcurrentRequests (on a servlet)
- PoolSize / FreePoolSize (on a connection pool)
- UsedMemory / FreeMemory (for the Java virtual machine)
- CPUUtilizationSinceLastMeasurement (for system CPU usage)

Managing alarms

Use the `Manage alarms` button to configure the alerts for a workspace. When you click this button a dialog opens which displays an overview of the current alarms. The manage alarms dialog has a button bar with the controls to create, read / update and delete alarms. Additionally there is also a button available to define an alarm blackout window. You can always view the alarms but editing is only allowed when you have at least editor privileges.

The following image shows the manage alarms dialog:



Name	Threshold	Timing	Severity	Action	Last fired
CPU Utilization	≥ 80	5 minute(s)	Medium	E-mail alert	Not Fired
JDBC Connection capacity	≤ 5	2 minute(s)	High	E-mail alert	Not Fired
Memory utilization	> 524.288	10 minute(s)	Low	E-mail alert	Not Fired
Twitter timeline performance	≥ 750	2 minute(s)	High	E-mail alert	Not Fired
Twitter timeline spike	≥ 5.000	Always	High	E-mail alert	Not Fired
Twitter timeline throughput	= 0	10 minute(s)	Medium	E-mail alert	Not Fired
Twitter timeline unavailable	Unavailable	60 second(s)	High	E-mail alert	Not Fired
WebContainer capacity	< 5	2 minute(s)	Medium	E-mail alert	Not Fired

Close

The green or gray bullet in the first column indicate if the alarm is enabled or disabled. The dialog also provides an overview of the most important alarm settings like the alarm threshold, the timing, severity and the action that is taken when the alarm is fired.

Creating and adjusting alarms

The manage alarms dialog contains a button bar with buttons to add, edit and delete an alarm. The buttons are only available when you have the required privileges. You can always double click on a row to view the detailed configuration of an alarm.

Use button `Add alarm` to create a new alarm. When you click on this button then an empty alarms dialog opens where you can specify the parameters which define the alarm. An alarm is defined for the workspace you are currently working in and is activated immediately after creation.

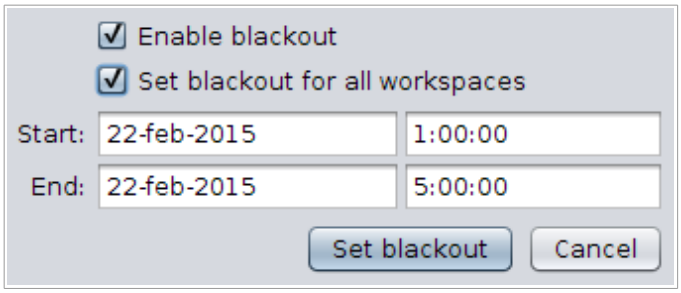
Use the button `Edit alarm` to modify an existing alarm. When you click on this button then the alarms dialog opens and the parameters for the alarm are loaded into the dialog. When you save the modified alarm this will reset current measurements for the alarm.

Use the button `Remove alarm` to delete the alarm. When you click on this button then PMIExplorer asks for confirmation to delete the alarm. Once alarm is deleted it stop immediately. It is not possible to undo the deletion of an alarm. Alarms that are delete by accident can only be recreated by adding a new alarm with the same parameters.

Defining an alarm blackout window

PMIExplorer allows you to define an alarm blackout window. During this time PMIExplorer will not fire alerts when thresholds are exceeded. You can use this function when you are performing planned maintenance to an application where you would like to suppress alarms.

Use the button `Define blackout` to configure the alarm blackout window. When you click on the button then the blackout dialog opens where you can configure the parameters for the blackout window:



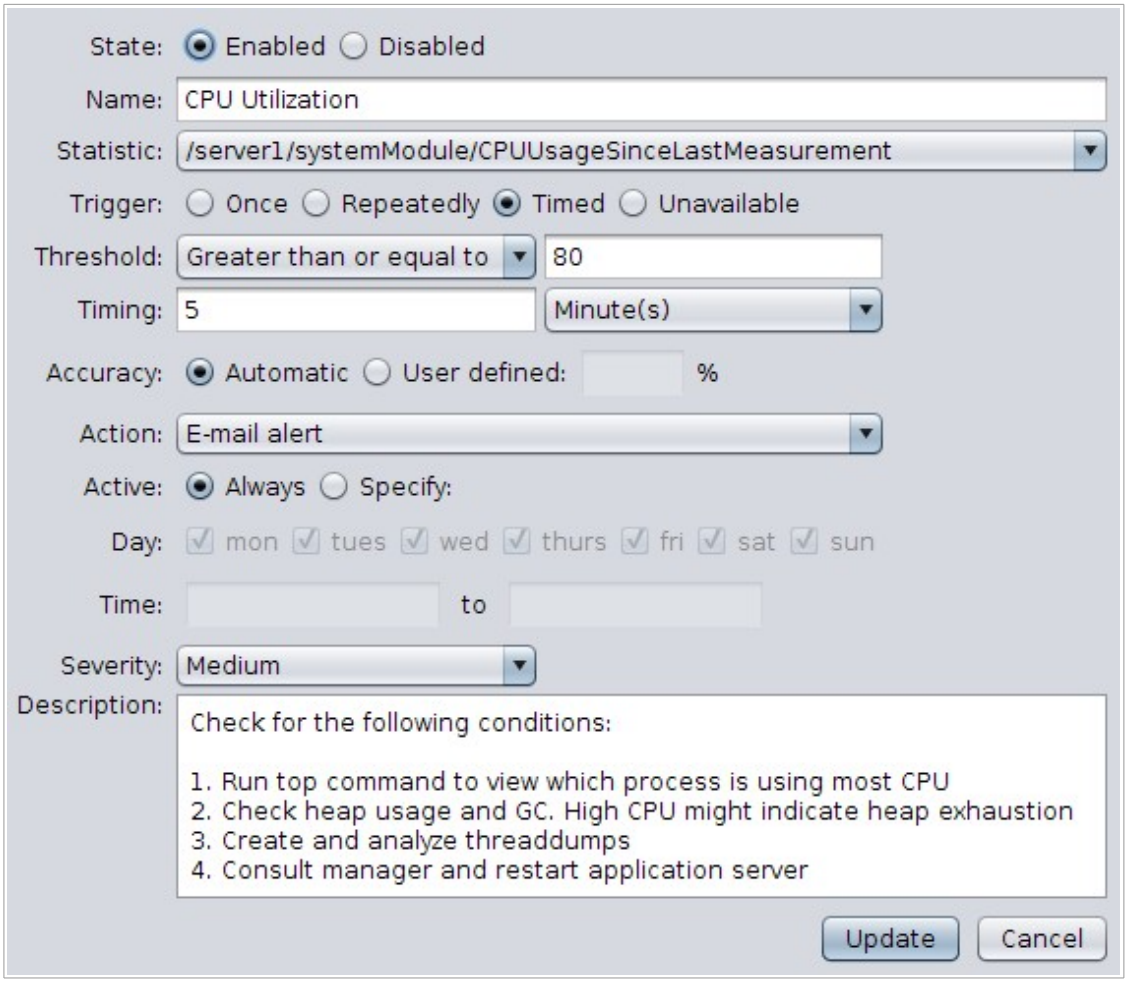
In the blackout dialog you need to enable the blackout window explicitly by checking the

`Enable blackout` box. Only then you can enter the `start` and `end` timestamps for the blackout window. It is possible to set the blackout globally for all workspaces, by checking the box. If you do not check the `Set blackout for all workspaces` then the blackout applies only to the current workspace.

When you define an alarm blackout window on the remote agent then this blackout window is persisted in the database and reloaded on agent restart. On the standalone client the blackout window is not persisted to the database.

Defining alerts

The alarm dialog contains a lot of options to define an alarm:


State: Enabled Disabled
Name: CPU Utilization
Statistic: /server1/systemModule/CPUUsageSinceLastMeasurement
Trigger: Once Repeatedly Timed Unavailable
Threshold: Greater than or equal to 80
Timing: 5 Minute(s)
Accuracy: Automatic User defined: %
Action: E-mail alert
Active: Always Specify:
Day: mon tues wed thurs fri sat sun
Time: to
Severity: Medium
Description:
Check for the following conditions:
1. Run top command to view which process is using most CPU
2. Check heap usage and GC. High CPU might indicate heap exhaustion
3. Create and analyze threaddumps
4. Consult manager and restart application server
Update
Cancel

General definition

To create an alarm you must define some general information about the alert. The `name` should give you a good indication what is going on with the system and what `statistic` is involved in the alarm. The `severity` provides an indicator to how important this alarm is.

The `statistic` refers to the recorded statistic. You can use all available statistics including:

- The actual recorded values

- The derived delta computations
- The aggregated values across multiple servers

The `state` of the alarm determines if the alarm is currently enabled. PMIExplorer will not fire an alert when the alarm is disabled (or when the alarm was fired in the blackout window).

The `description` field allows you to add additional information in the alarm. This information is also used the mail which is send to the operators when the alarm is fired. You can provide some instructions on how to proceed, when the alarm is fired.

Triggering

PMIExplorer has four `triggers` which can be applied to a statistic. For some of these triggers the `timing` and `accuracy` parameters also play a role when checking the alarm conditions:

Once:	An alarm with this trigger sends an alert when the threshold is exceeded in a measurement just once. The timing plays a role for this type and suppresses more alerts for the period indicated in the timing.
Repeatedly:	An alarm with this type always fires an alert when the threshold is exceeded. The timing parameter does not apply and is grayed you when this option is selected.
Timed:	A timed alarm is only fired when the measurements exceed the threshold during the period specified in the timing. The accuracy percentage determines how many measurements should have exceeded the threshold before the alarm is fired.
Unavailable:	The unavailable alert is fired when there are no measurements for a statistic during the period specified by the timing. The threshold does not apply for this type and is grayed out when this option is selected.

The `accuracy` percentage is applied to the timed and unavailable alarms. By default the accuracy compensates for the possibility that a measurement did not come through during the period specified in the timing, but it can also be used to refine performance alerts. The default accuracy rules define that an alarm is still fired under following conditions:

- 95% of measurements exceeded threshold with timing smaller than 5 minutes
- 96% of measurements exceeded threshold with timing between 5 and 10 minutes
- 97% of measurements exceeded threshold with timing between 10 and 30 minutes
- 98% of measurements exceeded threshold with timing between 30 minutes and 2 hours
- 99% of measurements exceeded threshold with timing greater than 2 hours

In general the `once` and `repeatedly` options are used to send alerts when you have huge spikes in response times or when you like to be notified when important functions in the application are called. In most cases alarms are defined based on timings like when the response times exceed the threshold for several minutes, available capacity drops during

several minutes or throughput is lower than what is normally expected during some time.

NOTE:

When you configure an alarm for when there is no throughput during a specified time frame you might need to override the accuracy and set it to include 100%, since the accuracy might otherwise send a false positive.

Threshold

The threshold defines the criterion which is continuously checked by PMIExplorer. The threshold has two inputs; the condition in a dropdown and the value in a text input. This value is compared with the value of the statistic where the condition is applies. If the outcome of that equation is true then the alarm is fired. The dropdown has the following options:

- Equal to
- Greater than or equal to
- Smaller than or equal to
- Greater than
- Smaller than

Timing

The timing defines the window that applies to an alarm trigger. For the `once` trigger the timing value means that further alarms should be suppressed when the threshold is exceeded in that window. For the `timed` and `unavailable` triggers the value means that measurements should exceed the threshold during this window before an alarm is fired.

The timing consists of two inputs; the value in a text input and the time unit in a dropdown. The following time units are available in the dropdown:

- Second(s)
- Minute(s)
- Hour(s)

Severity

The severity parameter can be used to indicate how serious the alert is. Different icons are used in PMIExplorer alerts based on this selection. The severity can be selected in a dropdown where the following options are available:

- High
- Medium
- Low

Action

The action in the alarm dialog determines what action is taken when an alarm is fired. PMIExplorer has two embedded alerting options which can be selected in the dropdown. There is also an extension point for alarms which allows to add additional alerting mechanisms to the alarm actions, like integration with your incident management system. The following options are available:

- Log events to alarms.log
- E-mail alert

NOTE:

The e-mail alert is only available when the notification properties have been configured in the PMIExplorer configuration file. Check the PMIExplorer log files for warnings and errors during startup for SMTP related messages if the e-mail alert option is not available in the alarm dialog.

Active

By default an alarm is always active but you have some options to override the default behavior. When an alarm is not active it will not generate alerts.

Use the checkboxes corresponding with the days of the week to define on which days an alert is active. You can disable alerting during weekends by not selecting the `sat` and `sun` options. When you specify the alarm window you can also configure the time wherein the alarm is active. If you specify from `09:00 to 17:00` and check all workdays of the would you will only receive notifications during office hours.

To enable nightly monitoring you can also specify the time to go past midnight. Specifying a time from 19:00 to 07:00 allows you to define a different threshold during for nightly monitoring. This might be useful for monitoring a system which different throughput depending on the time of the day.

Managing licenses

About the licensing system

PMIExplorer uses a licensing system with an activation key to verify your entitlement to use the software. This chapter explains how to manage your licenses after describing the details of the licensing system.

Activation period

The license key, which can be obtained from www.pmiexplorer.com, has a limited activation period. Depending on the type of license you can load the license into the software for only a few days after creating and downloading the license. When you load the license PMIExplorer will generate a system specific key (based on the machine hardware addresses).

During activation the `license.key` and `system.key` files are created in the PMIExplorer keys directory. These files are used to verify the integrity of the license when starting up PMIExplorer.

Type of license	Activation window
Beta	2 working days
Trail	2 working days
Standalone client	2 working days
Remote agent	3 working days
Enterprise	unlimited

NOTE:

Refer to terms and conditions on www.pmiexplorer.com for all terms and conditions and the end user license agreement. If for some reason you cannot import the license key within the specified activation window we offer some leniency. Please raise a support call to request extension of the activation window

Entitlements and renewing

The downloaded license file contains your entitlements to use the software. The remote agent and enterprise license have some configurable options. The remote agent license has a configurable amount of clients, which are allowed to connect to the remote agent and run standalone. The enterprise license includes the DNS domains in which the client, remote agent and WebSphere server must reside. You can upgrade your entitlement during the licensed period, but downgrading is only possible when the license period has expired.

30 days before the license expires the PMIExplorer software notifies you of the upcoming expiry. From that moment on it's possible to renew the license online, where you can also upgrade or downgrade the license. During the licensed period you can also upgrade your license (i.e. remote agent to enterprise), but downgrading (i.e. remote agent to standalone) is not possible.

License distribution service

In combination with the remote agent (remote agent or enterprise license) PMIExplorer includes a license distribution service. The goal of this license distribution service is automation of client license management. When a client without valid license connects to the remote agent, then the remote agent activates that client by distributing a license key to that client if the license permits it. For a remote agent license there is a limited number of clients that can be activated and when that amount is reached then the remote agent cannot activate more clients.

Updates to the license like upgrades or renewals are also pushed to clients when they connect and have a matching license key.

Logging

All action related to licensing are logged to the license.log file.

Importing the license key

Standalone clients

You can import the license for a standalone client using the `Manage license` dialog. When a valid license is loaded then this dialog shows the details of that license and otherwise the fields are empty. On this dialog there is a `Load...` or `Update...` button available which opens a file dialog. Now select your license key and press the `Load` button. This will activate your license and remove some restrictions within PMIExplorer.

Client in a remote agent topology

When the client is used as part of a remote agent- or enterprise license than there is no need to import the license key on the client. When the client connects to the agent then

the agent verifies the client license. If the license is not present then a license key is distributed from the agent to the client. When updates for the license are loaded on the remote agent, like renewals or updates, then these updates are also distributed to the clients.

Remote agent

For the remote agent the license key must be provided when the client connects to the remote agent. A file dialog opens automatically when the remote agent has no valid license key. Now select your license key and press the `Load` button. This will activate your license and remove some restrictions within the remote agent.

If the client did not have a valid license at this time it will automatically receive a license key from the remote agent. Administrator privileges are required to load the license on the remote agent.

Renewal, updating or upgrading

Renewing, updating or upgrading a license key works similar for the remote agent and standalone client. Using the `manage license` dialog you can view and update the current license. In the dialog the `Update...` button is available, which opens a file dialog. Select your new license key and press the `Load` button to activate that license in PMIExplorer.

In order to update the license on the remote agent you need to be connected to the remote agent, when updating the license. Updating the license on the remote agent requires that the user has administrator privileges.

Security

Authentication

Authenticating against LDAP

PMIExplorer security is only available for the remote agent topology, since data is shared amongst connected clients. Since the standalone client is intended to run on desktops only it does not make sense to authenticate in the client too.

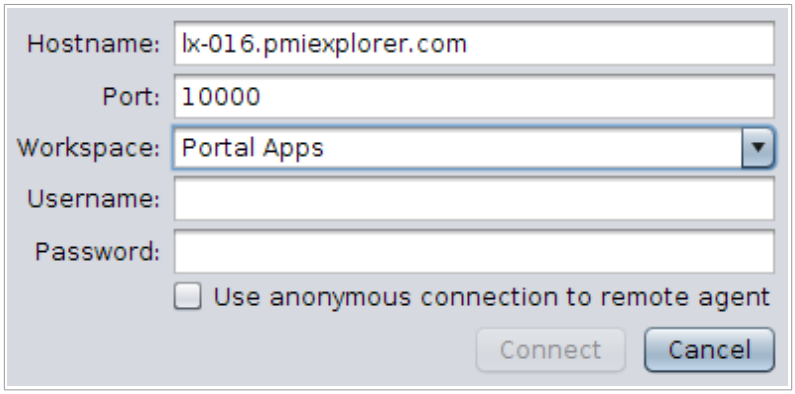
PMIExplorer does not have an embedded user management system, but can authenticate users using an LDAP based user repository. When a user authenticates then a bind is performed against the directory server. It is also possible to connect as an anonymous user.

You can configure your login name attribute in the PMIExplorer configuration by defining the user search filter. This filter is formatted like `(&(uid=%v)(objectclass=inetOrgPerson))` and must have at least one occurrence of `%v`. This search filter example returns all entries in the directory which are created with the objectclass `inetOrgPerson` and have an attribute `uid` which matches the username substitute variable. If the search filter returns multiple results then access is denied.

Please have a look at `LDAP Security` on page 24 for configuration details related to security.

Authenticate with the remote agent

When you connect to the remote agent you have to provide the connection details and your credentials as shown in the next picture.



It is possible to allow anonymous users to access the remote agent. In order to log in anonymous you need to check the `Use anonymous connection to remote agent` option. When security has not been configured for the remote agent, which is not recommended, then the provided credentials are ignored.

The presets in the connection dialog are loaded from the configuration file. Sensitive data, like the username and password, is encrypted with an AES cipher when it is transferred between the client and agent.

Special users

PMIExplorer has three special users which can be used while configuring access to the remote agent. These users are:

Anonymous user

Users who signed on to the remote agent with the `Use anonymous connection to remote agent` option are considered anonymous users.

Authenticated user

This special user applies to all users who have authenticated themselves against the directory server. This special user can be used to grant a role to everyone who is able to authenticate.

Primary administrator

This account is configured in the PMIExplorer configuration and must be used to configure the initial security. This account cannot be removed from the configuration. You might want to lock this account after you have set up the security.

Available roles

PMIExplorer has three roles within the application. Using these three roles you can grant authenticated or anonymous users permission to access certain functions in the program.

Reader

Users with the reader role have read only access to the remote agent. This role allows users to view real-time and recorded data from WebSphere endpoints (in multiple workspaces) and export this data to file. Users with reader permission also receive notifications when an alarm is fired. The details of the alert can be viewed through the manage alarms dialog, but the users are not allowed to make changes to the alarms.

Editor

The editor role also includes all permissions from the reader role. Users are also allowed to connect to WebSphere endpoints (and pause, resume, reconnect or remove endpoints) and configure the metrics to be recorded. Editors can configure the alerts and set up the real-time export to other systems. Permissions to create, load and unload monitoring profiles are also included in the editor role.

Administrator

Administrators have full access to the PMIExplorer. Besides all editor capabilities the additional functions which are accessible by these users include managing authorizations, manage the license and perform database maintenance task like collecting statistics and pruning old records.

Authorization

Authorizations in LDAP

Authorizations apply only to the remote agent topology. Once users have been

authenticated their authorizations are retrieved from the directory.

To determine authorizations PMIExplorer can use the group structure from your directory server. When an authenticated user is member of a group then you can use this group membership to grant a group of users a specific role in the application.

The lookup of groups in the directory is also performed with a search filter. This filter could look like `(&(cn=%v)(|(objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))` where the `%v` is replaced with the group name during the search. This example filter return all entries in the directory which are created with the objectclass `groupOfNames` or `groupOfUniqueNames` and have a `cn` attribute which matches the substitute variable. If the search filter return multiple results than this group is ignored during establishment of authorizations.

In the configuration you also have to define the group membership attributes. PMIExplorer performs a recursive group membership search based on the group membership attributes. If these attributes are found then an additional look-up is performed to find out if the user is a member of one of the nested groups. To prevent problems with circular group membership references PMIExplorer will not process more than 5 levels of nested groups.

Please have a look at `LDAP Security` on page `24` for configuration details related to security.

Configuring authorizations

Use the button `Configure security` to configure the authorizations. When you click on this button a dialog opens which gives access to the security configuration. In this dialog you can assign roles to users, special users or groups. The dialog allows you to add users or groups to the sections in the dialogs which correspond with the available roles; `administrator`, `editor` and `reader`.

PMIExplorer does not validate this input, since there is no tight coupling with the directory server. With the exception of the primary administrator all users, special users or groups can also be removed from a role.

When a change is made to security configuration than a notification is sent to all connected clients. This will trigger the client to re-validate the users permissions. Be careful not to remove yourself from the administrator role, since changes are effective immediately.

NOTE:

PMIExplorer recommends that you use a select group of editors and use the special anonymous and authenticated users only for the reader role.

Setting up the keystore

PMIExplorer uses a key database to save trusted (signer) certificates for connections to the WebSphere endpoints and secured directory servers. If you have WebSphere and / or company specific signers than you need to add those to the keystore.

To manage your keystore you could use a tool like ikeyman which is shipped as part of the IBM Java runtime environment.

NOTE:

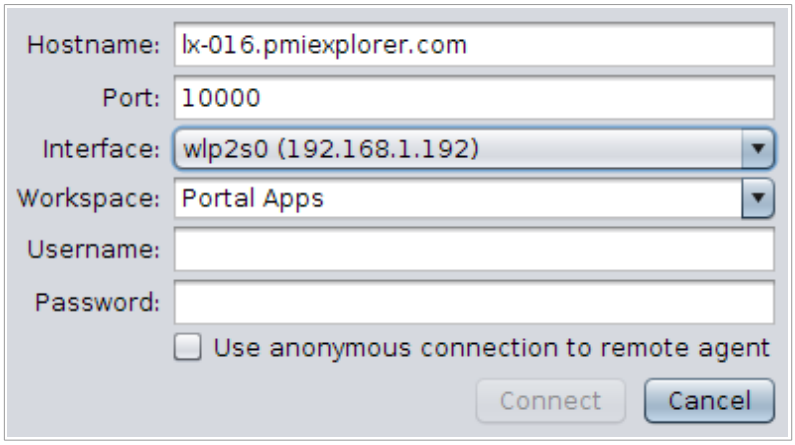
PMIExplorer also supports automatically acceptation of certificates when connecting to WebSphere endpoints or a LDAP directory server. To control the endpoints PMIExplorer connects to you could disable the automatic SSL certificate acceptation and add the certificates of the allows endpoints to the keystore. PMIExplorer is then unable to connect to other endpoints.

Remote agent

The remote agent component has been designed to keep monitoring in the background which removes the dependency on the client which often signs off at the end of a working day. The client and agent have the same monitoring capabilities, but only the agent supports multiple users accessing one workspace. For continues monitoring and alerting the agent is suited best and for troubleshooting sessions you could also use the standalone client.

Connecting to the remote agent

Use the button `Connect to remote agent` to establish a connection to the remote agent. When you click on this button a dialog opens where you can enter the connection details. The dialog also asks for a username and password, but also allows for anonymous connections. When the remote agent is not secured with LDAP security you must connect anonymously.



In the dialog you'll need to specify the connection details you receive from the PMIExplorer administrator. PMIExplorer and the libraries used to maintain the connection between agent and client did not register a port with ICANN and therefore your administrator is free to choose a port as long as it is not in use by another application.

The connection dialog requires that you to select the interface to be used in the connection on multi-homed systems (systems with multiple network interfaces). When you are connected to your organization over a VPN connection you are required to select that interface to access the PMIExplorer agent.

PMIExplorer can populate the host, port and workspaces values in the connection dialog with entries from the clients configuration file. This avoids problems for users might have forgotten to which endpoint they should connect.

Sensitive information like usernames and passwords are encrypted with an AES cipher while being transmitted between the client and agent.

Communication

RMI-IIOP

PMIExplorer uses the CORBA communication standards between the client and agent. Both the client and agent have implemented RMI-IIOP (RMI over IIOP) to exchange information. PMIExplorer uses the JacORB CORBA implementation because it support bidirectional communication between client and server, which means that the agent re-uses the connection from the client to send callbacks. By default RMI and CORBA do not re-use the connections in a callback architecture, but RMI and CORBA attempt to create a new connection from the agent back to the client, which complicates communication in firewalled or NAT environments.

The port which is opened by the agent actually runs a CORBA naming service which is used to initialize and negotiate the further connections between client and agent. This naming service is seen to the operating system as a separate process spawned from the agent.

Callbacks

PMIExplorer depends on callbacks from the agent to the client to keep communication efficient. This means that the client is not continuously polling the agent to get information, but when information is available on the agent it is relayed to the connected clients. The connected clients are essentially continuously downloading and processing information from the agent. In general the following information is relayed on an event basis:

Collector updates

When the collector records new data from the WebSphere endpoints this information is relayed to the client using a callback. This callback is then processed on the client which updates the panels and charts accordingly.

Alarms

When a recorded or computed value exceeds the configured threshold for an alarm an alert is relayed to the connected clients. This alert is pushed to the clients which update the internal state accordingly and show a pop-up on the desktop.

Monitoring configuration changes

Changes from users in a remote workspace are relayed to all other connected clients. When one client adds an endpoint or starts monitoring a metric then this action is relayed to all connected clients, which keeps their workspaces in sync.

Connectivity and keep alive

When the connection is established a small queue is created where messages for the client are temporarily buffered. In case of a connection hick-up this buffer is used to avoid loosing information. The client can compensate for a short connection failures which last some minutes. The information in the buffer is used by the client to catch up when the connection is re-established.

Both the client and agent have implemented a strategy to detect is the other party is still available. When the client has lost the connection it attempts to reconnect automatically. Then the agent is no longer able to relay information to the client it flushes the queues and removes the client registration from the callback mechanisms.

The alive checks also help keeping the connection open (alive) on routers and firewalls when there is no information to be relayed for a longer period of time.

The client also performs a keep alive to check for server availability. When it has lost the server for several pings it attempts to reconnect. When the client is on a laptop which goes into sleep or hibernation then the agent will disconnect that client automatically. The keep alive mechanism on the client also automatically reconnects to the remote agent when the laptop or desktop comes out of hibernation or sleep.

VPN and Firewall

PMIExplorer supports CORBA communication over VPN and through firewalls. Since there is additional negotiation taking place between the client and agent you might need to open up additional ports in your firewall configuration. The range of ports which is used in the negotiation can be influenced by setting two additional JacORB properties on the command line while starting the PMIExplorer agent:

```
-Djacorb.net.socket_factory.port.min=
-Djacorb.net.socket_factory.port.max=
```

Connection security

Encryption

The communication between the remote agent and the clients is not secured using SSL to avoid the overhead of encrypting all communications.

Sensitive data like usernames and passwords are protected nevertheless by encrypting the information on the client and decrypting the information on the agent. PMIExplorer uses an AES cipher to encrypt and decrypt the data that is transferred between client and agent.

Persistent monitoring

Monitoring settings

On the remote agent the monitoring configuration is persisted to the database. This enables PMIExplorer to automatically reload the monitoring settings and restart monitoring after a restart of the program. The persistent monitoring ensures that you can run your monitoring with minimal operational effort, also when performing maintenance on the operating system where PMIExplorer runs.

While persisting the monitoring settings also sensitive information like usernames and passwords are stored in the database. PMIExplorer encrypts this sensitive information with an AES cipher when it is stored in the database.

If you do not feel comfortable with saving this encrypted information then it is also possible to disable the persistent monitoring by setting the `pmi.agent.persistent` configuration property to `false`.

NOTE:

PMIExplorer does not monitor its own program. It is recommend to have external monitoring in place which checks if the remote agent is still running. The PMIExplorer program does not automatically restart

unless you have operating system specific services in place which launch the program after a system reboot. It is also possible to use high availability software, like Tivoli System Automation, to automatically relaunch the PMIExplorer program when it has stopped.

Reconnecting to WebSphere

To compensate for temporarily unavailability of the endpoints, which might occur due to maintenance on the WebSphere environment, PMIExplorer can automatically reconnect to a WebSphere endpoint. PMIExplorer checks the status every 5 minutes by default, when it reloads the endpoints to detect new servers and metrics. If an endpoint has become unavailable then PMIExplorer can attempt to reconnect to the endpoint if the configuration property `pmi.collector.reconnect` is set to `true` and the `pmi.collector.reload` interval is set to value greater than zero.

After a successful reconnect the monitoring profiles are also automatically applied again and monitoring is resumed normally. Depending on your alarm configuration an alert might be fired in the mean while if you have configured an alarm with an unavailability trigger for a statistic related to that endpoint.

When an endpoint becomes unavailable PMIExplorer distinguishes between unavailability due to authorization problems and unavailability due to other reasons. PMIExplorer does not attempt to reconnect when an authorization error has occurred because subsequent attempts to reconnect might lock this account.

NOTE:

To detect endpoint failures PMIExplorer recommends to define an alert on the jvmModule for each server on the endpoint. This alert will notify you when a JVM statistic for a server on the endpoint cannot be recorded anymore.